Aerodynamic Analysis and Optimisation of a Servo-Controlled Aileron

A Dissertation Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Science in Engineering
at the
University of Cape Town

Student Chris Day
Supervisor Chris Redelinghuys

September 2011

Abstract

With fuel prices rising while economies crash, and damage to the environment due to CO_2 emissions becoming ingrained in our global consciousness, the pressure for the airline industry to reduce its consumption of fuel is mounting. In a collaboration between a leading aircraft manufacturer and four South African universities, potentially more efficient technologies for aircraft roll control are being investigated. As purely manually-operated aircraft of a century ago became heavier, required stick forces for roll control were reduced using a counter-rotating tab to assist with aileron movement. With the availability of 21st century technology, it has been conjectured that smart materials could be employed to control the aileron deflections via the tab. This would replace bulky hydraulics with an electrically-actuated system, contributing to the airliner's continuous weight-loss programme.

This dissertation involves the aerodynamic analysis component of the collaboration project. Aerodynamic analyses were undertaken with a number of available open source programs such as XFoil, a vortex element panel method with coupled integral-method boundary layer; TSFoil, a non-linear potential-flow solver using the transonic small disturbance equations; and using CFD with the Spalart-Allmaras turbulence model. A range of configurations were tested using a double-flapped NACA 23012 airfoil, with aileron and tab deflected 4° and -4° respectively, and varying the angle of attack and freestream Mach number. It was found that only solutions of the full Navier-Stokes equations will give practically useful results for typical transonic cruise conditions.

Optimisation of the airfoil geometry – in terms of hinge locations as well as airfoil thickness and camber – was performed using the principles of Modern Design of Experiments. PABLO, a vortex panel method, was used to generate responses for lift and hinge moment coefficients, and results were compared using a full-factorial analysis and an L_{81} orthogonal array. The reduced array was seen to give accurate results even with just 11% of the data points of the full array. A merit function was evaluated and results suggest that a thick, symmetric airfoil performs best, while the aileron hinge should be moved forward and the tab hinge back, with the best performance found at high angles of attack. The applicability of the optimum found is limited by the inability of the panel method to accurately model transonic flow, but the analysis illustrates the usefulness of the efficient optimisation method when applied to a more computationally expensive analysis

like CFD.

Declaration

This dissertation is the result of my own work, except where explicit reference has been made to the work of others. No original material contained herein has been submitted previously, in whole or in part, for the award of any other academic degree or diploma.

Christopher Mark Day

Acknowledgments

I would like to especially thank the following people who have helped so kindly with the completion of this project.

Professor Christiaan Redelinghuys, for his seemingly inhuman patience, understanding and support.

The National Aeronautical Centre of Excellence (NACoE), for funding this research.

Mrs Paula Foley at the PGFO, for her remarkable observation and communication, going beyond the usual call of duty.

Mrs Susan Batho, for her extensive assistance with a large variety of administrative tasks.

Ms Marcelle Lorenzo, who provided much needed moral support, and for her wonderfully infectious good nature.

Table of Contents

Abstract i

Declaration iii

Acknowledgments iv

Table of Contents v

List of Figures xi

List of Tables xviii

Glossary of Terms xx

List of Symbols xxiv

1. Introduction 1

Part I – Aerodynamic Analysis 4

2. General Theoretical Treatment of Airfoils 5

2.1 Introduction . . . 5
2.2 Modelling Airfoils . . . 5
2.3 Forces and Moments on an Airfoil . . . 6
2.4 Calculating the Forces and Moments . . . 6
2.5 Dimensionless Coefficients . . . 7
2.6 Boundary Conditions . . . 8

3. Theory Of Fluid Flow 9

3.1 Introduction . . . 9
3.2 Origin of Fluid Flow Equations . . . 10
3.3 Potential Flow . . . 11
3.3.1 Compressible Potential Flow . . . 11
3.3.2 Steady Incompressible Potential Flow . . . 12
3.3.3 Compressibility Corrections for Incompressible Flow . . . 12
3.3.4 Transonic Small Disturbance Theory . . . 13
3.3.5 The Kutta Condition . . . 15

3.4 Boundary Layer Theory . . . 15
3.4.1 Boundary Layer Development . . . 16
3.4.2 Pressure Distribution In The Boundary Layer . . . 17
3.4.3 Integral Method Boundary-Layer Equations . . . 17
3.4.4 Thwaites' Method for Laminar Boundary Layers . . . 19
3.4.5 Entrainment Methods for Turbulent Boundary Layers . . . 20
3.4.6 Predicting Drag . . . 21
3.5 Transition . . . 22
3.5.1 Mechanisms of Transition . . . 22
3.5.2 Fully Turbulent Assumption for Transport Aircraft . . . 24
3.5.3 Michel's Criterion For Predicting Turbulence . . . 24
3.5.4 e^N Methods For Predicting Turbulence . . . 24
3.6 Separation . . . 26
3.6.1 Laminar Separation Bubbles . . . 26
3.7 Turbulence Modelling . . . 27
3.7.1 The Spalart-Allmaras Turbulence Model . . . 28
3.7.2 Mesh Size Requirement . . . 30

4. Aerodynamic Software 31
4.1 Introduction . . . 31
4.2 Available Aerodynamic Software . . . 31
4.3 Programs Chosen . . . 32
4.4 XFoil . . . 32
4.4.1 Analysis . . . 33
4.4.2 Running XFoil . . . 35
4.5 TSFoil . . . 37
4.5.1 Analysis . . . 37
4.5.2 Running TSFoil . . . 38
4.5.3 Preliminary Results . . . 39
4.6 PABLO . . . 41
4.6.1 Features of PABLO . . . 41
4.6.2 Analysis . . . 43
4.7 CFD . . . 43
4.7.1 Introduction . . . 43
4.7.2 Setting Up – I: The Mesh . . . 43
4.7.3 Setting Up – II: The Solver . . . 47
4.7.4 Validation . . . 49

5. Aerodynamic Analyses 52
5.1 Introduction . . . 52
5.2 Simulations Performed . . . 52
5.3 Application of the Kármán-Tsien Compressibility Correction . . . 53

5.4 Comparison: Linear Potential Flow and with Viscous Boundary-Layer . . . 54
5.5 Comparison: Non-linear and Linear Potential Flow . . . 59
5.6 Comparison: Navier-Stokes and Linear Potential Flow with Viscous Boundary Layer . . . 64
5.7 Comparison: Navier-Stokes and Non-linear Potential Flow . . . 66
5.8 Discussion Of Results . . . 71
5.9 CFD Analyses . . . 72
5.9.1 Introduction . . . 72
5.9.2 Contour Plots for Mach-Sweep Simulations . . . 73
5.9.3 Contour Plots for α-Sweep Simulations . . . 79
5.9.4 A Closer Look At Shock Capturing . . . 86
5.9.5 A Closer Look At Shock/Boundary-Layer Interaction . . . 89

6. Modifications to PABLO **93**
6.1 `pablo.m` . . . 94
6.2 `create_profile.m` . . . 95
6.3 `distribute_x.m` . . . 95
6.4 `rotate_hinge.m` . . . 96
6.5 `vortex.m` . . . 98
6.6 `lift.m` . . . 99
6.7 `moment.m` . . . 99
6.8 Verification with XFoil . . . 100
6.9 Pablo Variable Explanation . . . 101
6.9.1 `airfoil` . . . 101
6.9.2 `alpha` . . . 102
6.9.3 `c_h_a`, `c_h_t`, `c_h` . . . 102
6.9.4 `c_l` . . . 102
6.9.5 `C_p` . . . 102
6.9.6 `delta_a`, `delta_t`, `delta` . . . 102
6.9.7 `nbpo2` . . . 102
6.9.8 `x_a`, `y_a`, `x_t`, `y_t`, `x_h`, `y_h` . . . 102
6.9.9 `x` . . . 102
6.9.10 `xl`, `yl`, `xu`, `yu` . . . 103
6.9.11 `z`, `za` . . . 103

Part II – Optimisation **104**

7. Optimisation Introduction **105**

8. Optimisation Theory **106**
8.1 Basic Principles of Modern Design of Experiments . . . 106

8.2 Types of Experimental Arrays . . . 107
8.3 Factor Effects . . . 109
8.4 Interactions . . . 111
8.5 Calculating Degrees of Freedom . . . 111
8.6 Confounding: Linear Graphs and Interaction Tables . . . 112
8.7 Building an Empirical Model . . . 114
8.8 Analysis of Variance . . . 116
8.9 Hypothesis Testing . . . 117
8.10 Tabulating ANOVA Results . . . 119
8.11 Standard Mathematical Models . . . 119
8.12 Model Refining Methods . . . 120
8.13 Scaled Residuals . . . 121
8.13.1 Standardised Residuals . . . 121
8.13.2 Studentised Residuals . . . 121
8.13.3 PRESS Residuals and the PRESS Statistic . . . 122
8.14 Residual Analysis . . . 122
8.15 Model Evaluation . . . 124
8.15.1 The Multiple Correlation Coefficient, R^2 . . . 124
8.15.2 The C_p Statistic . . . 125
8.15.3 Adequate Precision . . . 125
8.16 Power Transformations on the Response . . . 125

9. Optimisation Methodology 127
9.1 Introduction . . . 127
9.2 Problem Description . . . 127
9.2.1 The Merit Function . . . 127
9.2.2 Factors and Responses . . . 128
9.3 Full-Factorial Analysis . . . 131
9.3.1 Setting Up The Experiment . . . 131
9.3.2 Analysing The Output . . . 134
9.3.3 Resulting Equations . . . 141
9.4 Fractional-Factorial Analysis . . . 143
9.4.1 Selecting an Orthogonal Array . . . 143
9.4.2 Results of Factor Column Assignment . . . 143
9.4.3 Analysis in Design Expert . . . 144
9.4.4 Resulting Equations . . . 148

10. Final Analysis And Results 150
10.1 Model Terms And Coefficients . . . 150
10.2 Factor Effects . . . 154
10.3 Interactions . . . 159
10.4 Merit Function Analysis . . . 167
10.4.1 Calculating The Merit Function . . . 167

10.4.2 Configurations With No Solution . . . 168
10.5 Merit Function Factor Effects . . . 171
10.6 L_{81} Accuracy Analysis . . . 172
10.6.1 L_{81} Residual analysis . . . 172
10.6.2 Full-Factorial Plots . . . 178

11. Conclusions **183**
11.1 Aerodynamic Analyses . . . 183
11.2 Optimisation . . . 183

References **185**

Appendix **191**

Appendix A. XFoil Ancillary Code **A-1**

Appendix B. TSFoil I/O Files **B-1**
B.1 Input File `n23012.inp` . . . B-1
B.2 Output File `output.out` . . . B-2
B.3 Output File `fort.1` . . . B-11

Appendix C. Calculation of Properties for Cruising Airliner **C-1**
C.1 Chord . . . C-1
C.2 Altitude . . . C-1
C.3 Temperature, Pressure and Density . . . C-2
C.4 Viscosity . . . C-2
C.5 Speed of Sound . . . C-2
C.6 Velocity . . . C-2
C.7 Reynolds Number . . . C-3

Appendix D. CFD Output **D-1**
D.1 Lift and Drag History Plots . . . D-1
D.1.1 α-sweep at $M = 0.5$ . . . D-2
D.1.2 α-sweep at $M = 0.8$ . . . D-9
D.1.3 Mach-sweep at $\alpha = 3°$ . . . D-16
D.2 Wall $y+$ Values and Example Plots . . . D-23

Appendix E. PABLO `.m` Files **E-1**
E.1 `pablo.m` . . . E-1
E.2 `create_profile.m` . . . E-1
E.3 `distribute_x.m` . . . E-3
E.4 `rotate_hinge.m` . . . E-4
E.5 `vortex.m` . . . E-5
E.6 `lift.m` . . . E-7

E.7 `moment.m` E-8

Appendix F. Matlab Optimisation Code **F-1**

F.1 `generate_output_full.m` F-1
F.2 `generate_output_L81.m` F-2
F.3 `assign_factors.m` F-4
F.4 `interaction_table.m` F-5
F.5 `check_confounding.m` F-7
F.6 `L81.m` F-7

Appendix G. List of Standard Orthogonal Arrays **G-1**

Appendix H. L_{81} Example Analysis **H-1**

Appendix I. Merit Function Analysis Code **I-1**

I.1 `merit_function_L81.m` I-1
I.2 Deciding which δ_a to use I-4

Appendix J. Matlab Code for Interaction Plots **J-1**

Appendix K. Matlab Code for Statistical Analysis **K-1**

List of Figures

1 Variation of $C_{p_{\text{crit}}}$ with M_∞ xxi
2 Cosine spacing applied during airfoil discretisation xxi

1.1 A Flettner rudder, or servo tab 1
1.2 Block diagram of the CoSICS project 2

2.1 Airfoil nomenclature 5
2.2 Pressure and shear stress on an airfoil surface 6
2.3 Equivalent force and moment systems on an airfoil 6
2.4 Integration of pressure and shear stress distribution 7

3.1 Hierarchy of mathematical models used to analyse fluid flow . . 9
3.2 Complexity vs. different classes of computer 10
3.3 Elementary flows 13
3.4 Comparison of compressibility corrections 14
3.5 Allowable flow regimes for application of the TSD equations . . 14
3.6 Airfoil with no condition of flow over the trailing edge 15
3.7 Airfoil with application of the Kutta condition 15
3.8 Laminar boundary layer along a flat plate at zero incidence . . 16
3.9 Typical boundary layer development over a subsonic airfoil . . . 16
3.10 Displacement of the outer flow due to the boundary layer 18
3.11 Physical interpretation of momentum thickness 18
3.12 Laminar to turbulent transition 22
3.13 Attachment line for a swept wing 23
3.14 The formation of cross-flow instabilities due to sweep angle Λ . 23
3.15 Transition on a Boeing 737 slat at cruise conditions 24
3.16 Stability diagram for e^N method 25
3.17 Streamlines within the boundary layer around the separation point 26
3.18 The anatomy of a laminar separation bubble 27
3.19 Evidence of laminar separation bubbles 27

4.1 Daedalus human powered aircraft in flight 33
4.2 XFoil output of C_p vs $\frac{x}{c}$ for viscous analysis 34
4.3 XFoil command prompt at program start 36
4.4 Plot of XFoil airfoil profile node distribution 36
4.5 XFoil `.ps` output of result shown in Fig. 4.2 37

4.6 Control volumes for TSFoil inviscid drag calculations 38
4.7 Opening screen of TSFoil 38
4.8 TSFoil command prompt output warning 39
4.9 TSFoil contours of the velocity field 40
4.10 TSFoil C_p distribution 40
4.11 TSFoil velocity distribution 41
4.12 PABLO inviscid solutions using different methods 42
4.13 PABLO viscous solution showing extra output 42
4.14 View of complete CFD mesh 44
4.15 View of interior of CFD mesh 45
4.16 View of the innermost mesh block 45
4.17 Close-up view of the mesh at the leading edge 46
4.18 Close-up view of the mesh at the trailing edge 46
4.19 Fluent menu pane example 47
4.20 Quadrilateral mesh used by Fluent for validation purposes . . . 49
4.21 Mach distribution as calculated by Fluent 50
4.22 Mach distribution as calculated by the author 50
4.23 Comparison of pressure coefficient 51
4.24 Comparison of skin friction coefficient 51

5.1 Kármán-Tsien compressibility correction 53
5.2 XFoil viscous vs inviscid comparison, $\alpha = 3°$, $M = 0.1$ 54
5.3 XFoil viscous vs inviscid comparison, $\alpha = 3°$, $M = 0.2$ 54
5.4 XFoil viscous vs inviscid comparison, $\alpha = 3°$, $M = 0.3$ 55
5.5 XFoil viscous vs inviscid comparison, $\alpha = 3°$, $M = 0.4$ 55
5.6 XFoil viscous vs inviscid comparison, $\alpha = 3°$, $M = 0.5$ 55
5.7 XFoil viscous vs inviscid comparison, $\alpha = 3°$, $M = 0.6$ 56
5.8 XFoil viscous vs inviscid comparison, $M = 0.3$, $\alpha = -10°$ 56
5.9 XFoil viscous vs inviscid comparison, $M = 0.3$, $\alpha = -5°$ 57
5.10 XFoil viscous vs inviscid comparison, $M = 0.3$, $\alpha = 0°$ 57
5.11 XFoil viscous vs inviscid comparison, $M = 0.3$, $\alpha = 5°$ 58
5.12 XFoil viscous vs inviscid comparison, $M = 0.3$, $\alpha = 10°$ 58
5.13 XFoil viscous vs inviscid comparison, $M = 0.3$, $\alpha = 15°$ 59
5.14 TSFoil vs XFoil inviscid formulation for $M = 0.5$, $\alpha = 3°$ 60
5.15 TSFoil vs XFoil inviscid formulation for $M = 0.6$, $\alpha = 3°$ 60
5.16 TSFoil vs XFoil inviscid formulation for $M = 0.7$, $\alpha = 3°$ 60
5.17 TSFoil vs XFoil inviscid formulation for $M = 0.8$, $\alpha = 3°$ 61
5.18 TSFoil vs XFoil inviscid formulation, $M = 0.5$, $\alpha = -9°$. 61
5.19 TSFoil vs XFoil inviscid formulation, $M = 0.5$, $\alpha = -6°$. 62
5.20 TSFoil vs XFoil inviscid formulation, $M = 0.5$, $\alpha = -3°$. 62
5.21 TSFoil vs XFoil inviscid formulation, $M = 0.5$, $\alpha = 0°$. 62
5.22 TSFoil vs XFoil inviscid formulation, $M = 0.5$, $\alpha = 3°$. 63

5.23 TSFoil vs XFoil inviscid formulation, $M = 0.5$, $\alpha = 6°$. 63
5.24 TSFoil vs XFoil inviscid formulation, $M = 0.5$, $\alpha = 9°$. 63
5.25 XFoil vs CFD C_p distribution for $M_\infty = 0.4$ and $\alpha = 3°$ 64
5.26 XFoil vs CFD C_p distribution for $M_\infty = 0.5$ and $\alpha = 3°$ 65
5.27 XFoil vs CFD C_p distribution for $M_\infty = 0.6$ and $\alpha = 3°$ 65
5.28 XFoil vs CFD C_p distribution for $M_\infty = 0.7$ and $\alpha = 3°$ 66
5.29 TSFoil vs CFD C_p distribution for $\alpha = 3°$, $M_\infty = 0.5$ 67
5.30 TSFoil vs CFD C_p distribution for $\alpha = 3°$, $M_\infty = 0.6$ 67
5.31 TSFoil vs CFD C_p distribution for $\alpha = 3°$, $M_\infty = 0.7$ 68
5.32 TSFoil vs CFD C_p distribution for $\alpha = 3°$, $M_\infty = 0.8$ 68
5.33 TSFoil vs CFD C_p distribution for $\alpha = 3°$, $M_\infty = 0.9$ 68
5.34 TSFoil vs CFD C_p distribution for $\alpha = -6°$, $M_\infty = 0.5$ 69
5.35 TSFoil vs CFD C_p distribution for $\alpha = -3°$, $M_\infty = 0.5$ 69
5.36 TSFoil vs CFD C_p distribution for $\alpha = 0°$, $M_\infty = 0.5$ 69
5.37 TSFoil vs CFD C_p distribution for $\alpha = 3°$, $M_\infty = 0.5$ 70
5.38 TSFoil vs CFD C_p distribution for $\alpha = 6°$, $M_\infty = 0.5$ 70
5.39 C_p distribution over NACA 23012 at $\alpha = 3°$, $M_\infty = 0.4$ 73
5.40 Mach number distribution over NACA 23012 at $\alpha = 3°$, $M_\infty = 0.4$ 73
5.41 Viscosity distribution over NACA 23012 at $\alpha = 3°$, $M_\infty = 0.4$. 73
5.42 C_p distribution over NACA 23012 at $\alpha = 3°$, $M_\infty = 0.5$ 74
5.43 Mach number distribution over NACA 23012 at $\alpha = 3°$, $M_\infty = 0.5$ 74
5.44 Viscosity distribution over NACA 23012 at $\alpha = 3°$, $M_\infty = 0.5$. 74
5.45 C_p distribution over NACA 23012 at $\alpha = 3°$, $M_\infty = 0.6$ 75
5.46 Mach number distribution over NACA 23012 at $\alpha = 3°$, $M_\infty = 0.6$ 75
5.47 Viscosity distribution over NACA 23012 at $\alpha = 3°$, $M_\infty = 0.6$. 75
5.48 C_p distribution over NACA 23012 at $\alpha = 3°$, $M_\infty = 0.7$ 76
5.49 Mach number distribution over NACA 23012 at $\alpha = 3°$, $M_\infty = 0.7$ 76
5.50 Viscosity distribution over NACA 23012 at $\alpha = 3°$, $M_\infty = 0.7$. 76
5.51 C_p distribution over NACA 23012 at $\alpha = 3°$, $M_\infty = 0.8$ 77
5.52 Mach number distribution over NACA 23012 at $\alpha = 3°$, $M_\infty = 0.8$ 77
5.53 Viscosity distribution over NACA 23012 at $\alpha = 3°$, $M_\infty = 0.8$. 77
5.54 C_p distribution over NACA 23012 at $\alpha = 3°$, $M_\infty = 0.9$ 78
5.55 Mach number distribution over NACA 23012 at $\alpha = 3°$, $M_\infty = 0.9$ 78
5.56 Viscosity distribution over NACA 23012 at $\alpha = 3°$, $M_\infty = 0.9$. 78
5.57 C_p distribution over NACA 23012 at $M_\infty = 0.8$, $\alpha = -9°$ 79
5.58 Mach no. distribution over NACA 23012 at $M_\infty = 0.8$, $\alpha = -9°$ 79
5.59 Viscosity distribution over NACA 23012 at $M_\infty = 0.8$, $\alpha = -9°$ 79
5.60 C_p distribution over NACA 23012 at $M_\infty = 0.8$, $\alpha = -6°$ 80
5.61 Mach no. distribution over NACA 23012 at $M_\infty = 0.8$, $\alpha = -6°$ 80
5.62 Viscosity distribution over NACA 23012 at $M_\infty = 0.8$, $\alpha = -6°$ 80
5.63 C_p distribution over NACA 23012 at $M_\infty = 0.8$, $\alpha = -3°$ 81
5.64 Mach no. distribution over NACA 23012 at $M_\infty = 0.8$, $\alpha = -3°$ 81

5.65 Viscosity distribution over NACA 23012 at $M_\infty = 0.8$, $\alpha = -3°$ 81
5.66 C_p distribution over NACA 23012 at $M_\infty = 0.8$, $\alpha = 0°$ 82
5.67 Mach number distribution over NACA 23012 at $M_\infty = 0.8$, $\alpha = 0°$ 82
5.68 Viscosity distribution over NACA 23012 at $M_\infty = 0.8$, $\alpha = 0°$. 82
5.69 C_p distribution over NACA 23012 at $M_\infty = 0.8$, $\alpha = 3°$ 83
5.70 Mach number distribution over NACA 23012 at $M_\infty = 0.8$, $\alpha = 3°$ 83
5.71 Viscosity distribution over NACA 23012 at $M_\infty = 0.8$, $\alpha = 3°$. 83
5.72 C_p distribution over NACA 23012 at $M_\infty = 0.8$, $\alpha = 6°$ 84
5.73 Mach number distribution over NACA 23012 at $M_\infty = 0.8$, $\alpha = 6°$ 84
5.74 Viscosity distribution over NACA 23012 at $M_\infty = 0.8$, $\alpha = 6°$. 84
5.75 C_p distribution over NACA 23012 at $M_\infty = 0.8$, $\alpha = 9°$ 85
5.76 Mach number distribution over NACA 23012 at $M_\infty = 0.8$, $\alpha = 9°$ 85
5.77 Viscosity distribution over NACA 23012 at $M_\infty = 0.8$, $\alpha = 9°$. 85
5.78 Qualitative characteristics of flow through a normal shock wave 86
5.79 Qualitative depiction of a property that decreases over a shock 87
5.80 Density, ρ, in kg/m^3 for $M_\infty = 0.8$, $\alpha = 0°$ 87
5.81 Static temperature, T, in K for $M_\infty = 0.8$, $\alpha = 0°$ 88
5.82 Enthalpy, h, in J/kg for $M_\infty = 0.8$, $\alpha = 0°$ 88
5.83 Velocity vectors in the boundary layer 89
5.84 Contours of effective viscosity in the boundary layer 89
5.85 Velocity vectors showing separation 90
5.86 Contours of viscosity showing separation 90
5.87 Close up of airfoil at the base of the shock for $M_\infty = 0.8$ 91
5.88 Close up of airfoil at the base of the shock for $M_\infty = 0.7$ 91
5.89 Close up of airfoil at the base of the shock for $M_\infty = 0.6$ 92

6.1 PABLO C_p distributions vs XFoil for a NACA 4412 airfoil . . . 93
6.2 PABLO node spacing for a 120-panel NACA 23012 airfoil . . . 96
6.3 Flap deflection procedure: Step 1 97
6.4 Flap deflection procedure: Step 2 97
6.5 Flap deflection procedure: Step 3 97
6.6 Flap deflection procedure: Step 3 in detail 98
6.7 Verification of PABLO - C_L 100
6.8 Verification of PABLO - C_H 101
6.9 Explanation of hinge locations 103

8.1 MDOE P-Diagram 106
8.2 Example of how factor effects may be qualitatively analysed . . 110
8.3 Analysis of interactions 111
8.4 Interaction between A and B leading to possible confounding. . 113
8.5 Linear graphs for L_8 (2^7) orthogonal array 114
8.6 Geometric interpretation of the sum of squares identity 116
8.7 Shape of normal distribution for histogram plots 123

8.8 Normal probability plot example 123
8.9 Example of a well-behaved residual scatter plot 124

9.1 The two cases for the merit function 128
9.2 New design in Design Expert using the Historical Data option . . 131
9.3 Definition of "categoric" factors in Design Expert 132
9.4 Definition of the responses in Design Expert 132
9.5 Full-factorial data from Matlab inserted into Design Expert . . 133
9.6 Model selection in Design Expert 133
9.7 Dialog box to choose to preserve model hierarchy 134
9.8 Design Expert ANOVA output – Page 1 135
9.9 Design Expert ANOVA output – Page 2 136
9.10 Design Expert ANOVA output – Page 3 137
9.11 Normal plot of e_i for full-factorial analysis – $\alpha = 0.1$ 139
9.12 Normal plot of e_i for full-factorial analysis – $\alpha = 0.05$ 139
9.13 e_i vs Predicted plot for full-factorial analysis – $\alpha = 0.1$ 140
9.14 e_i vs Predicted plot for full-factorial analysis – $\alpha = 0.05$ 140
9.15 Normal plot of residuals for fractional-factorial analysis – $\alpha = 0.1$ 146
9.16 Normal plot of residuals for fractional-factorial analysis – $\alpha = 0.05$ 146
9.17 e_i vs Predicted plot for fractional-factorial analysis – $\alpha = 0.1$ . . 147
9.18 e_i vs Predicted plot for fractional-factorial analysis – $\alpha = 0.05$. 147

10.1 Factor effects for C_L using full-factorial analysis 155
10.2 Factor effects for C_L using fractional-factorial analysis 155
10.3 Factor effects for C_{Ha} using full-factorial analysis 156
10.4 Factor effects for C_{Ha} using fractional-factorial analysis 156
10.5 Factor effects for C_{H_t} using full-factorial analysis 157
10.6 Factor effects for C_{H_t} using fractional-factorial analysis 157
10.7 Interaction between α and δ_a 160
10.8 Interaction between α and δ_t 160
10.9 Interaction between α and x_a 161
10.10 Interaction between α and x_t 161
10.11 Interaction between α and NACA 162
10.12 Interaction between δ_a and δ_t 162
10.13 Interaction between δ_a and x_a 163
10.14 Interaction between δ_a and x_t 163
10.15 Interaction between δ_a and NACA 164
10.16 Interaction between δ_t and x_a 164
10.17 Interaction between δ_t and x_t 165
10.18 Interaction between δ_t and NACA 165
10.19 Interaction between x_a and x_t 166
10.20 Interaction between x_a and NACA 166
10.21 Interaction between x_t and NACA 167

10.22 C_{H_a} vs. δ_a for 00012 using full-factorial analysis 168
10.23 C_{H_a} vs. δ_a for 23012 using full-factorial analysis 169
10.24 C_{H_a} vs. δ_a for 00012 using fractional-factorial analysis 169
10.25 C_{H_a} vs. δ_a for 23012 using fractional-factorial analysis 170
10.26 Merit function factor effects for full-factorial analysis 171
10.27 Merit function factor effects for fractional-factorial analysis . . . 171
10.28 Scatter plot of d_i vs C_L 172
10.29 Normal % probability distribution of d_i vs C_L 173
10.30 Histogram of e_i distribution for C_L 174
10.31 Scatter plot of d_i vs C_{H_a} 174
10.32 Normal % probability distribution of d_i vs C_{H_a} 175
10.33 Histogram of e_i distribution for C_{H_a} 176
10.34 Scatter plot of d_i vs C_{H_t} 176
10.35 Normal % probability distribution of d_i vs C_{H_t} 177
10.36 Histogram of e_i distribution for C_{H_t} 178
10.37 Scatter plot of d_i vs C_L 178
10.38 Normal % probability distribution of d_i vs C_L 179
10.39 Histogram of e_i distribution for C_L 179
10.40 Scatter plot of d_i vs C_{H_a} 180
10.41 Normal % probability distribution of d_i vs C_{H_a} 180
10.42 Histogram of e_i distribution for C_{H_a} 181
10.43 Scatter plot of d_i vs C_{H_t} 181
10.44 Normal % probability distribution of d_i vs C_{H_t} 182
10.45 Histogram of e_i distribution for C_{H_t} 182

C.1 Airbus datasheet excerpt showing aileron dimensions C-1

D.1 Lift coefficient history for $\alpha = -9°$, $M_\infty = 0.5$ D-2
D.2 Drag coefficient history for $\alpha = -9°$, $M_\infty = 0.5$ D-2
D.3 Lift coefficient history for $\alpha = -6°$, $M_\infty = 0.5$ D-3
D.4 Drag coefficient history for $\alpha = -6°$, $M_\infty = 0.5$ D-3
D.5 Lift coefficient history for $\alpha = -3°$, $M_\infty = 0.5$ D-4
D.6 Drag coefficient history for $\alpha = -3°$, $M_\infty = 0.5$ D-4
D.7 Lift coefficient history for $\alpha = 0°$, $M_\infty = 0.5$ D-5
D.8 Drag coefficient history for $\alpha = 0°$, $M_\infty = 0.5$ D-5
D.9 Lift coefficient history for $\alpha = 3°$, $M_\infty = 0.5$ D-6
D.10 Drag coefficient history for $\alpha = 3°$, $M_\infty = 0.5$ D-6
D.11 Lift coefficient history for $\alpha = 6°$, $M_\infty = 0.5$ D-7
D.12 Drag coefficient history for $\alpha = 6°$, $M_\infty = 0.5$ D-7
D.13 Lift coefficient history for $\alpha = 9°$, $M_\infty = 0.5$ D-8
D.14 Drag coefficient history for $\alpha = 9°$, $M_\infty = 0.5$ D-8
D.15 Lift coefficient history for $\alpha = -9°$, $M_\infty = 0.8$ D-9
D.16 Drag coefficient history for $\alpha = -9°$, $M_\infty = 0.8$ D-9

D.17 Lift coefficient history for $\alpha = -6°$, $M_\infty = 0.8$ D-10
D.18 Drag coefficient history for $\alpha = -6°$, $M_\infty = 0.8$ D-10
D.19 Lift coefficient history for $\alpha = -3°$, $M_\infty = 0.8$ D-11
D.20 Drag coefficient history for $\alpha = -3°$, $M_\infty = 0.8$ D-11
D.21 Lift coefficient history for $\alpha = 0°$, $M_\infty = 0.8$ D-12
D.22 Drag coefficient history for $\alpha = 0°$, $M_\infty = 0.8$ D-12
D.23 Lift coefficient history for $\alpha = 3°$, $M_\infty = 0.8$ D-13
D.24 Drag coefficient history for $\alpha = 3°$, $M_\infty = 0.8$ D-13
D.25 Lift coefficient history for $\alpha = 6°$, $M_\infty = 0.8$ D-14
D.26 Drag coefficient history for $\alpha = 6°$, $M_\infty = 0.8$ D-14
D.27 Lift coefficient history for $\alpha = 9°$, $M_\infty = 0.8$ D-15
D.28 Drag coefficient history for $\alpha = 9°$, $M_\infty = 0.8$ D-15
D.29 Lift coefficient history for $M_\infty = 0.4$, $\alpha = 3°$ D-16
D.30 Drag coefficient history for $M_\infty = 0.4$, $\alpha = 3°$ D-16
D.31 Lift coefficient history for $M_\infty = 0.5$, $\alpha = 3°$ D-17
D.32 Drag coefficient history for $M_\infty = 0.5$, $\alpha = 3°$ D-17
D.33 Lift coefficient history for $M_\infty = 0.6$, $\alpha = 3°$ D-18
D.34 Drag coefficient history for $M_\infty = 0.6$, $\alpha = 3°$ D-18
D.35 Lift coefficient history for $M_\infty = 0.7$, $\alpha = 3°$ D-19
D.36 Drag coefficient history for $M_\infty = 0.7$, $\alpha = 3°$ D-19
D.37 Lift coefficient history for $M_\infty = 0.8$, $\alpha = 3°$ D-20
D.38 Drag coefficient history for $M_\infty = 0.8$, $\alpha = 3°$ D-20
D.39 Lift coefficient history for $M_\infty = 0.9$, $\alpha = 3°$ D-21
D.40 Drag coefficient history for $M_\infty = 0.9$, $\alpha = 3°$ D-21
D.41 Wall $y+$ for $M = 0.4$, $\alpha = 3°$ D-24
D.42 Wall $y+$ for $M = 0.8$, $\alpha = 3°$ D-24
D.43 Wall $y+$ for $\alpha = -9°$, $M = 0.8$ D-25
D.44 Wall $y+$ for $\alpha = 9°$, $M = 0.8$ D-25

H.1 Final equation as calculated by Design Expert H-4

I.1 Example of C_{H_a} vs. δ_a for a single configuration I-4

List of Tables

4.1 Typical N_{crit} values for XFoil's e^N transition calculation 35
4.2 Allowable ranges of M and α in TSFoil 39
4.3 Fluid properties for Fluent validation 49

5.1 Simulations performed for comparative purposes 53
5.2 Summary of TSFoil weak-shock violation warnings 66
5.3 Trends for the properties given in Fig. 5.78 86

8.1 Experimental runs for a full-factorial programme 107
8.2 Test matrix for an experimental programme. 108
8.3 Experimental runs for a fractional-factorial programme 109
8.4 Calculating minimum runs from degrees of freedom. 112
8.5 Taguchi $L_8(2^7)$ array 113
8.6 2^3 full factorial array 113
8.7 Interaction table for L_8 (2^7) orthogonal array 114
8.8 Typical layout of ANOVA table 119
8.9 Box-Cox power transforms 126

9.1 Factors used for the opimisation procedure 129
9.2 Control factor levels for MDOE analysis 129
9.3 Model characteristics for C_L full-factorial analysis 138
9.4 Model characteristics for C_{H_a} full-factorial analysis 138
9.5 Model characteristics for C_{H_t} full-factorial analysis 138
9.6 Calculation of the degrees of freedom for the L_{81} analysis 143
9.7 $L_{81}(3^{40})$ column assignment 143
9.8 Results of hand-checking with published interaction table . . . 144
9.9 Model characteristics for C_L fractional-factorial analysis 144
9.10 Model characteristics for C_{H_a} fractional-factorial analysis 145
9.11 Model characteristics for C_{H_a} fractional-factorial analysis 145

10.1 Comparison of model terms and coefficients for C_L analysis . . 151
10.2 Comparison of model terms and coefficients for C_{H_a} analysis . . 152
10.3 Comparison of model terms and coefficients for C_{H_t} analysis . . 153
10.4 Percentage difference in average values for factor effects – C_L . 155
10.5 Percentage difference in average values for factor effects – C_{H_a} . 156
10.6 Percentage difference in average values for factor effects – C_{H_t} . 157

10.7 Average values for factor effects calculations – C_L 158
10.8 Average values for factor effects calculations – C_{H_a} 158
10.9 Average values for factor effects calculations – C_{H_t} 158
10.10 Outliers of C_L analysis 172
10.11 Outliers of C_{H_a} analysis 174
10.12 Outliers of C_{H_t} analysis 177

B.1 TSFoil input variables B-1

C.1 Standard atmosphere values at 10 000 m C-2
C.2 Reynolds number for range of Mach numbers C-3

D.1 Final Values of C_L and C_D for α-sweep at Mach 0.5 . . . . . D-22
D.2 Final Values of C_L and C_D for α-sweep at Mach 0.8 . . . . . D-22
D.3 Final Values of C_L and C_D for Mach-sweep at $\alpha = 3°$ D-22
D.4 Minimum and maximum values of $y+$ for α-sweep at $M = 0.5$. D-23
D.5 Minimum and maximum values of $y+$ for α-sweep at $M = 0.8$. D-23
D.6 Minimum and maximum values of $y+$ for Mach-sweep at $\alpha = 3°$ D-23

G.1 Standard orthogonal arrays G-1

H.1 Test array for example analysis H-1

Glossary of Terms

Aerodynamic Centre

The aerodynamic centre of a planform is the point about which the pitching moment is constant with a variation in angle of attack.

Angle of Attack, α

The angle of attack is the angle between the chord, c, and the freestream velocity, V_∞.

Boundary Layer

A thin layer of flow near a body where viscous effects are dominant, as distinct from the "outer flow" where viscous effects are assumed negligible.

Chord, c

The chord is a straight line from the leading edge to the trailing edge of an airfoil.

Critical Mach Number, $\mathbf{M}_{\text{crit}}$

The critical Mach number, M_{crit}, is the value of the freestream Mach number where the flow initially becomes transonic. M_{crit} can vary depending on the airfoil geometry and the angle of attack, but is typically around $M_\infty = 0.7$.

Critical Pressure Coefficient, $\mathbf{C}_{\mathbf{p}_{\text{crit}}}$

The critical pressure coefficient is the value of the pressure coefficient for which the flow is at sonic velocity. The derivation of $C_{p_{\text{crit}}}$ can be found in most books on compressible flow, and the equation below is taken from §3.5 of [1]. The relationship between M_∞ and $C_{p_{\text{crit}}}$ is plotted in Fig. 1 showing subsonic and supersonic regions of the flow.

$$C_{p_{\text{crit}}} = \frac{2}{\gamma M_\infty^2}\left[\left(\frac{2}{\gamma+1} + \frac{\gamma-1}{\gamma+1}M_\infty^2\right)^{\frac{\gamma}{\gamma-1}} - 1\right]$$

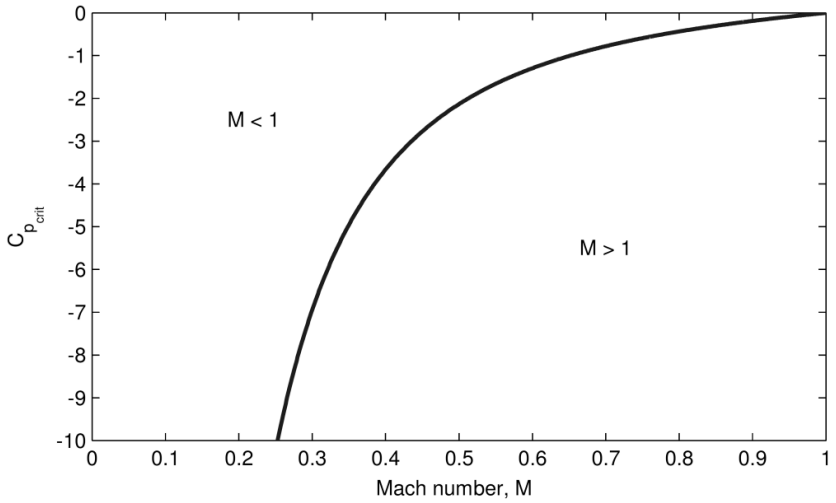


Figure 1: Variation of $C_{p_{\text{crit}}}$ with M_∞

Cosine Spacing

Cosine spacing is a means of spatially distributing x-coordinate points on an airfoil for computational analyses. For θ distributed uniformly in the range $0 < \theta < \pi$, a vector is created having points generated according to the equation

$$\xi = \frac{c}{2}(1 - \cos\theta)$$

It can be seen in Fig. 2 when theta is varied uniformly, ξ will have a more dense distribution of points at the leading and trailing edges of the airfoil.

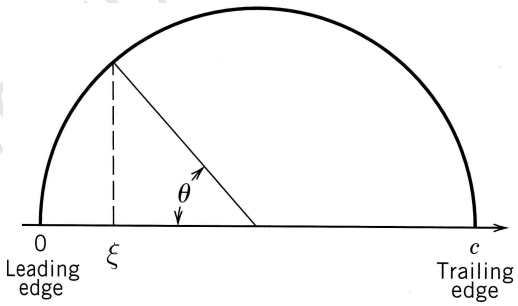


Figure 2: Cosine spacing applied during airfoil discretisation [2]

Effective Viscosity

Prandtl considered an analogy between the transfer of viscous momentum due to exchange of molecules in adjacent layers in laminar flow, and turbulent momentum transfer due to movement of eddies over relatively large distances. This is known as mixing length or eddy length. Thus, analogous to Newton's law of viscosity, Prandtl proposed

$$\tau_{yx,t} = -\mu_t \frac{\mathrm{d}\mathbf{v}_x}{\mathrm{d}y}$$

where $\tau_{yx,t}$ is the turbulent shear stress, and μ_t is the turbulent viscosity generated as a result of movement of eddies over relatively large distances (eddy length) compared to the movement of molecules over the mean free path in laminar flow. The term *effective viscosity*, μ_{eff}, is used for the sum of molecular viscosity, μ, and turbulent viscosity, μ_t. In a turbulent flow, μ is negligible in comparison to μ_t, and hence the turbulent and effective viscosity values are almost the same. [3]

Farfield

In aerodynamic computer simulations the farfield usually refers to the outer boundary of the control volume where the effect of the object being modelled has a negligible effect on the flow.

Hierarchy

A regression model is said to be hierarchical if the inclusion of higher-order terms such as interactions or second order terms requires the inclusion of the lower-order terms of which they are made up. As an example when building a model the first-order term x_1 may be found to be insignificant, but if the interaction x_1x_3 is significant then x_1 must be included.

The principle of hierarchy is generally considered to be a reasonable model-building practice when fitting polynomials [4].

No-slip Condition

This is a criterion applied to all viscous analyses, where the velocity immediately adjacent to the wall is considered to be zero.

Reynolds-Averaged Navier-Stokes Equations

The Navier-Stokes equations are often decomposed into the Reynolds-Averaged Navier-Stokes (RANS) equations in order to simulate practical engineering flows. The assumption (known as the Reynolds decomposition) behind the RANS equations is that the time-dependent turbulent velocity fluctuations can be separated from the mean flow velocity. This transform then introduces a set of unknowns called the Reynolds stresses, which are functions of the velocity fluctuations, and which require a turbulence model to produce a closed system of soluble equations. [5]

Shape Memory Alloys

Shape memory alloys form part of a larger group of materials of "smart structures" which have the ability to change mechanical characteristics such as stiffness and shape in response to changes in temperature, electric or magnetic field. Shape memory alloys exhibit a diffusionless martensitic transformation when cooled from

the higher-temperature austenitic state, and this property is used to incite changes in geometry with the application of an electric current. [6]

List of Symbols

The symbols listed here are the default values of those used within this dissertation, unless otherwise stated in the text.

Roman Letters

a	sonic velocity	m/s
c	chord length	m
C	dimensionless coefficient (when used with a subscript)	–
c_v	heat capacity at constant volume	J/K
D	drag	N
e	residual	–
E	energy	J
F	factor	
h	height	m
H	enthalpy	J/kg
H	boundary layer shape factor	–
k	coefficient of thermal conductivity	–
k	number of regressors in a regression model	–
l	length	m
L	lift force	N
m	mean value	
M	Mach number	–
M	merit function	
M	moment	Nm
n	number of runs in a regression analysis	–
N	factor used in e^N methods	–
p	pressure	Pa
p	number of terms in a regression model	–
q	dynamic pressure	Pa
s	arc length (usually along airfoil surface)	m
S	surface	–
t	time	s
T	temperature	$^\circ C$ or K
u, v, w	component of velocity in the x, y, z direction	m/s
$\boldsymbol{v}$, $\mathbf{V}$ or V	total velocity	m/s
x, y, z	Cartesian coordinates	m

$\mathbf{X}$	design matrix	–
y	response	–

Greek Letters

α	angle of attack	degrees
α	significance level	–
β	coefficient of regression term	–
δ	flap deflection angle	degrees
δ	boundary layer thickness	m
ϵ, $\boldsymbol{\varepsilon}$ or ε	error term	–
ϕ	velocity potential	m^2/s
γ or Γ	circulation	m/s
γ	heat capacity ratio	–
η	response	–
Λ	sweep angle	degrees
μ	absolute viscosty	$Pa\,s$
ν	kinematic viscosity	m^2/s
θ	angle	degrees
θ	momentum thickness	m
ρ	density	kg/m^3
σ	stress	Pa
σ^2	standard deviation	–
τ	shear stress	Pa
ω	rotational velocity	rad/s
Ω	control volume	–
ξ	vorticity	rad/s
ψ	incompressible stream function	m^2/s

Subscripts

a	aileron
crit	critical, when a change in condition occurs
D	drag
e	edge (of boundary layer)
H	hinge moment
l	lower surface
L	lift
n	normal
p	pressure
r	circular coordinate radial direction
t	tab
u	upper surface
w	property at the wall boundary

x, y, z	displacement w.r.t direction
x, y, z	differentiation w.r.t direction
xx, yy, zz	differentiation w.r.t direction
θ	circular coordinate rotational direction
0	total, or stagnation (for pressure)
∞	freestream value

Superscripts

$'$	per unit span

Mathematical symbols

$\hat{\mathbf{i}}$	unit vector in the x-direction
$\hat{\mathbf{j}}$	unit vector in the y-direction
$\mathbf{n}$	normal unit vector
Δ	total change in given quantity
$\times$	cross product
δ_{ij}	Kronecker delta
$\times$	multiplication
$\neq$	not equal to
$\approx$	approximately equal to
$\equiv$	equivalent to
∇	gradient operator
∇^2	Laplacian operator

Acronyms and Abbreviations

ANOVA	Analysis of Variance
GUI	Graphical User Interface
LE	Leading Edge
MDOE	Modern Design of Experiments
NACA	National Advisory Committee for Aeronautics
Pr	Prandtl number
RAE	Royal Aircraft Establishment
RANS	Reynolds-averaged Navier-Stokes
Re	Reynolds number
§	Section
TE	Trailing Edge
TSD	Transonic Small Disturbance

Constants

γ	ratio of specific heats for air	1.4
R	universal gas constant	8.31451 $J/K \cdot mol$

Dimensionless Coefficients

C_D	coefficient of drag (per unit length)
C_H	hinge moment coefficient (per unit length)
C_L	coefficient of lift (per unit length)
C_L	pressure coefficient (per unit length)

Other Variables

$\overline{y}$	mean value of response
$\widehat{y}$	predicted response
c_v	specific heat at constant volume

1 Introduction

Twenty-five airlines stopped operating in the first half of 2008 in the most concentrated occurrence of bankruptcy within the industry of any time in its history. The International Air Transport Association (IATA) places much of the blame on the rising cost of fuel, which has resulted in fuel costs increasing from 13 to 35 percent of an airliner's annual budget [7]. In addition, the airline industry has committed itself to a 50% reduction in fuel consumption and CO_2 emissions by 2012, and believe that 10% can be gained from aerodynamic refinements and a further 5% by weight saving. Other gains would come from advances in engine and fuel technology [8].

The benefits of fuel saving are therefore both environmental and economical, increasing the importance and urgency of making any progress towards this goal. This is the main focus of the current project – to save fuel through one of many aerodynamic refinements and weight saving strategies taking place around the aircraft.

Smart-actuated, tab assisted ailerons

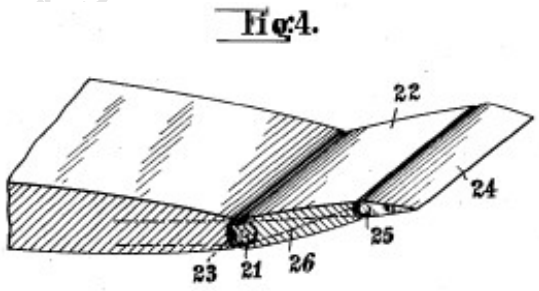


Figure 1.1: A Flettner rudder, or servo tab [9]

Tab-assisted control surfaces, like that seen in Fig. 1.1, were in operation as far back as during the first world war. This so-called Flettner rudder had the tab connected directly to the pilot's control, so that when it was deflected the main control surface moved in the opposite direction [10]. There is a smaller force on the tab, which reduces the pilot's input, but the longer moment arm from the *aerodynamic centre* of the tab to the main hinge more than compensates for the opposing aerodynamic force on the main control surface.

Actuators made from *shape memory alloys* (SMAs) have recently been pro-

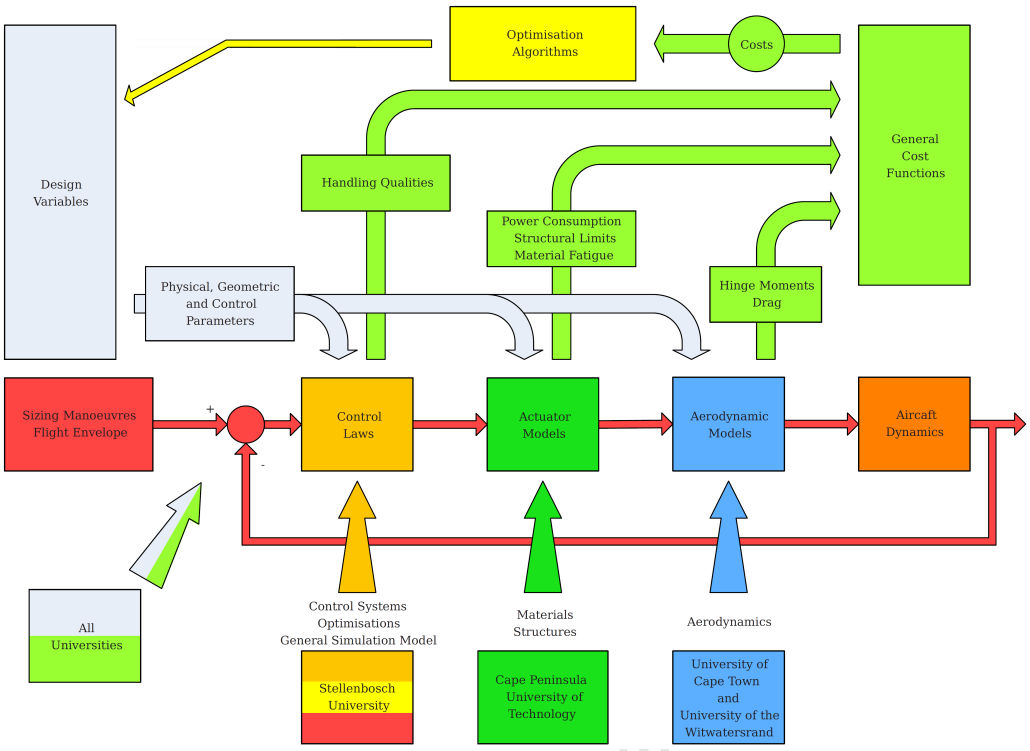


Figure 1.2: Block diagram of the CoSICS project showing the aerodynamic model (blue) which is to be provided during the course of this project

posed for application in the aeronautics industry [6]. They generally offer higher mechanical energy density and lower power consumption compared with similar hydraulic counterparts [11]. A major aircraft manufacturer has therefore initiated a collaborative effort between four South African universities in order to explore the possibility of replacing the hydraulic aileron actuators with SMA actuators in a servo tab configuration.

Project Aims

The role of the University of Cape Town in the collaboration is to provide an aerodynamic model which is to be integrated into the flight path analysis performed by Stellenbosch University using Matlab/Simulink. An overview of the project, titled "Control Surfaces in Confined Spaces", or simply CoSICS, is shown in Fig. 1.2. The aerodynamic component encompassed by this project is shown in blue as the "Aerodynamic Module", and must satisfy a set of inputs and outputs to enable successful integration into the CoSICS project.

In addition to the aerodynamic modelling, an optimisation will also be performed using the principles of modern design of experiments (MDOE). These have been successfully employed at NASA Langley to solve a variety of problems [12–14], most of which have been related to wind tunnel testing [15–17]. The approach will be examined using a linear, potential flow analysis to test the efficacy of the optimisation procedure by comparison of the heavily-reduced data set with a full data

set.

Dissertation Contents

The dissertation consists of two parts. The first contains the aerodynamic analysis, which begins with a review of the theory involved. Aerodynamic software is then reviewed, and some programs are chosen to use. An investigation of the results they give in accordance with their mathematical complexity follows.

The second part contains the optimisation analysis, which again starts with a theory section. The methodology used to achieve the results is then given, along with some intermediate results, followed by the final results of the optimisation procedure. The dissertation concludes with some remarks on the methods used and conclusions are drawn about the results acquired.

The appendices contain some Matlab code used, as well as data that would otherwise break the flow of the document.

Formatting Notes

Terms that are *emphasised*, such as the two examples found in this introduction, can be found in the Glossary of Terms.

Referencing within a sentence refers to the text in that sentence only. A reference appearing outside of a sentence refers to the preceding sentences of that paragraph, or a selection thereof where obvious. And referencing at the end of a paragraph that appears justified to the right of the text refers to the entire paragraph. All unoriginal pictures have been referenced in the corresponding caption.

Part I

Aerodynamic Analysis

2 General Theoretical Treatment of Airfoils

2.1 Introduction

This section describes the theoretical treatment of airfoil aerodynamics, in only as much detail as is necessary to understand the software algorithms used. Described here are the orientation of an airfoil in a coordinate system, forces and moments and their calculation, dimensionless coefficients, and boundary conditions applied during analysis. Chapter 3 will endeavour to describe the specific theories used to solve the actual airflow parameters.

2.2 Modelling Airfoils

An airfoil is shown in its conventional orientation on a Cartesian axis system in Fig. 2.1, where the leading edge (LE) is placed at the origin, with the trailing edge (TE) on the x-axis at $x = c$. For the vertical direction y and z are used interchangeably. For a dimensionless analysis the x-axis is normalised using the chord length so that $c = 1$, and the freestream velocity, V_∞, is taken as unity. The angle of the freestream velocity varies with angle of attack, α, measured positive as shown in Fig. 2.1. Flap deflections (not shown here) are given by δ and are measured positive downwards.

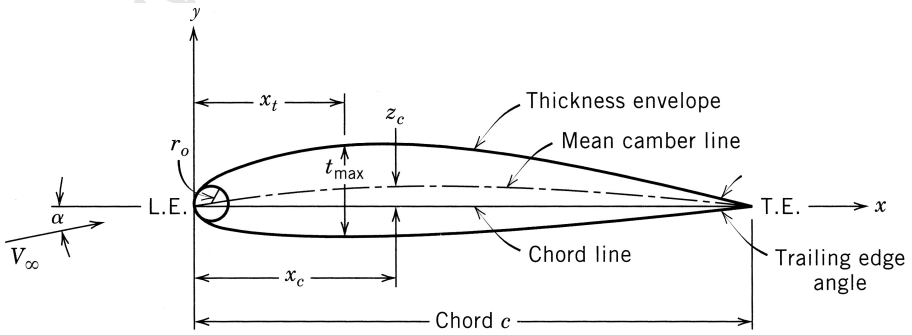


Figure 2.1: Airfoil nomenclature [18]

For computational purposes, when an airfoil coordinate file needs to be handled, a common convention – and the one used here – is for the airfoil surface coordinates to start at the trailing edge, move continuously along the upper then the lower surfaces, ending again at the trailing edge.

2.3 Forces and Moments on an Airfoil

The forces on an airfoil can be seen to be due to the pressure, p, acting normal to the airfoil surface, and the shear stress, τ, which acts tangentially. These are distributed over the surface of the airfoil, and are shown for a point in Fig. 2.2. The net effect of the pressure and stress distributions can be combined for convenience into a single force and moment system. This system can be given around various points, as seen in Fig. 2.3.

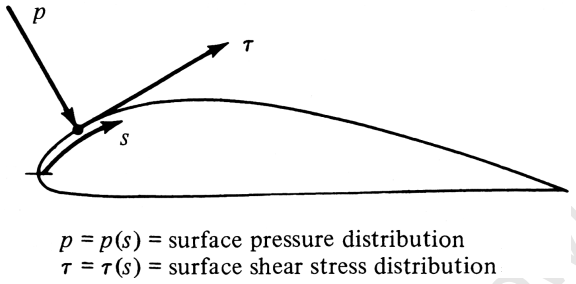


Figure 2.2: Pressure and shear stress on an airfoil surface [2]

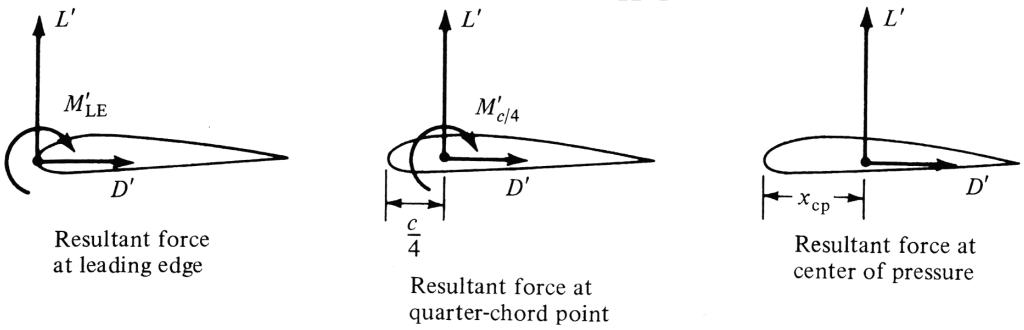


Figure 2.3: Equivalent force and moment systems on an airfoil [2]

The net force is conventionally broken up into components of lift and drag, by definition acting perpendicular and parallel to V_∞ respectively. Both α and δ can then be seen to be measured positively when causing a net increase in lift. Moments are also measured with a similar philosophy, and are positive when clockwise, so that a positive moment acts on an airfoil to cause an increase in lift.

2.4 Calculating the Forces and Moments

Aerodynamic solvers give the pressure and sometimes shear stress distribution along the airfoil surface. Integration can then be performed over the surface to find the net force, which is then deconstructed to give the lift and drag. The moment can also be calculated in a similar way by integrating the cross product of the resultant force and the lever arm from the moment axis over that part of the surface to be considered. In two dimensions, referring to Fig. 2.4, the moment

about the leading edge due to the forces on the surface can be given by [2]

$$M'_{LE} = \int_{LE}^{TE} \left[(p_u \cos\theta + \tau_u \sin\theta)\, x + (-p_u \sin\theta - \tau_u \cos\theta)\, y \right] ds_u + \int_{LE}^{TE} \left[(-p_l \cos\theta + \tau_l \sin\theta)\, x - (p_l \sin\theta - \tau_l \cos\theta)\, y \right] ds_l$$

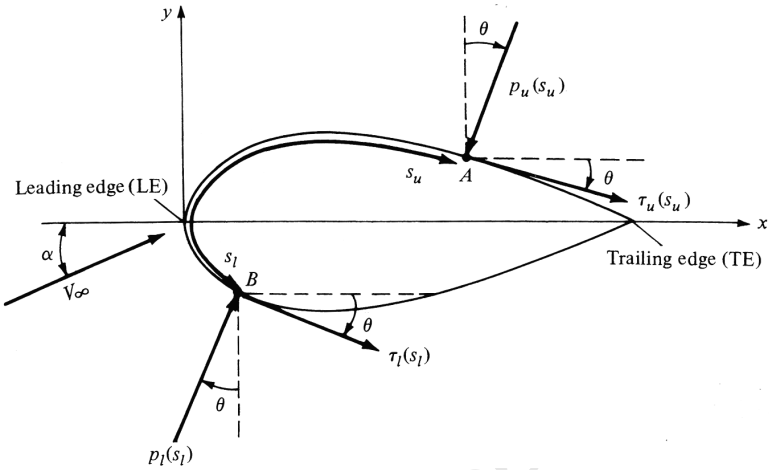


Figure 2.4: Integration of pressure and shear stress distribution [2]

2.5 Dimensionless Coefficients

All forces and moments acting on an airfoil can be converted to dimensionless coefficients in order to broaden the validity of the results. By non-dimensionalisation, results are applicable for any airfoil of similar shape and other flowfield similarity parameters, depending on the analysis used. For example, when a viscous result is undertaken, if the Reynolds and Mach numbers are the same, then any airfoil of similar shape[1] will develop a similar flowfield pattern.

The forces and moments are non-dimensionalised using the chord length and the freestream dynamic pressure, q_∞, where $q_\infty = \frac{1}{2}\rho V^2$. The lift-, drag-, and moment-coefficients per unit span are reported here and used throughout this dissertation using capitals, and are given by

$$C_L = \frac{L'}{q_\infty c}, \quad C_D = \frac{D'}{q_\infty c}, \text{ and } \quad C_H = \frac{M'}{q_\infty c}$$

The pressure can also be non-dimensionalised in a similar way, and in this case the pressure coefficient is defined as

$$C_p \equiv \frac{p - p_\infty}{q_\infty}$$

where p is the local pressure, p_∞ is the freestream pressure, and q_∞ is the dynamic

[1] Here "shape" includes the orientation of the airfoil in the freestream flow, given by α.

pressure given by

$$q_\infty 2\rho V^2$$

2.6 Boundary Conditions

Boundary conditions can vary depending on the analysis applied as well as the physical layout of the problem. Throughout this project unbounded flow was used, where the so-called *farfield* boundary conditions are given by

$$V_{\infty,\,x} = V_\infty \cos\alpha\hat{\mathbf{i}}$$
$$V_{\infty,\,y} = V_\infty \sin\alpha\hat{\mathbf{j}}$$

The wall boundary condition for an inviscid analysis gives that velocity at the surface of the airfoil is assumed to be a streamline of the flow, so that

$$V_n = 0$$

For viscous analyses implementation of the *no-slip condition* at the wall results in

$$\mathbf{V} = 0$$

3 Theory Of Fluid Flow

3.1 Introduction

The behaviour of fluid flow is most fully described using the conservation and constitutive laws, which can be cast in integral or differential form depending on the type of analysis undertaken. Here these are briefly presented in §3.2 following Jameson's summary in [19]. The resulting (so-called Navier-Stokes) equations are inherently non-linear and therefore very difficult to solve, unless geometric or other simplifications are made.

Of the several simplifications that can be made to the Navier-Stokes equations, each one will cause certain terms to be omitted. Figure 3.1 shows a hierarchy of computational models at different levels of simplicity, where the lower tier, marked *I*, reduces the equations so that they are linear and therefore easily soluble. By simply leaving out terms some associated features of the flow will not be modeled. But even so, the solutions of these reduced equations will be valid for certain ranges of flow regimes depending on the terms omitted. So although our computational capabilities have increased somewhat, the methods introduced in the 60s are still applicable, and in use, today.

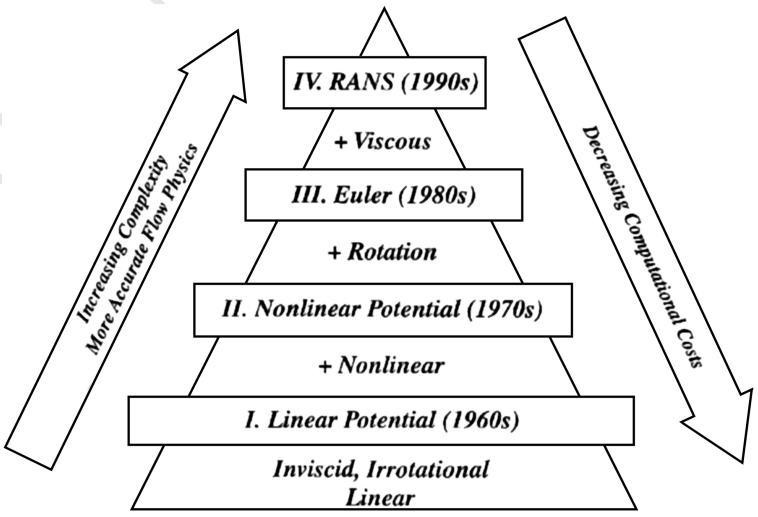


Figure 3.1: Hierarchy of mathematical models used to analyse fluid flow, with their period of origin and the complexities introduced [19]

Even at our current maximum computational capacity the Navier-Stokes equations undergo Reynolds- and/or time-averaging, and small-scale effects are ap-

proximated, for example by turbulence models which capture vorticity orders of magnitude smaller that the grid size used. Computational costs for fluid analysis vary greatly, with Fig. 3.2 showing what kind of analyses are feasible according to the computational speed available. Modern desktop PCs now produce peak speeds of around 50 gigaflops [20], but sustained speeds might only be 10% of this depending on architecture variables like memory size. Here the trade-off between accuracy and cost must be evaluated depending on the application and desired outcomes. For example, during the initial design phase, evaluation of different models requires a fast turn-around time, and potential theories may be sufficient, while in the final design stages high-fidelity analyses are a necessity.

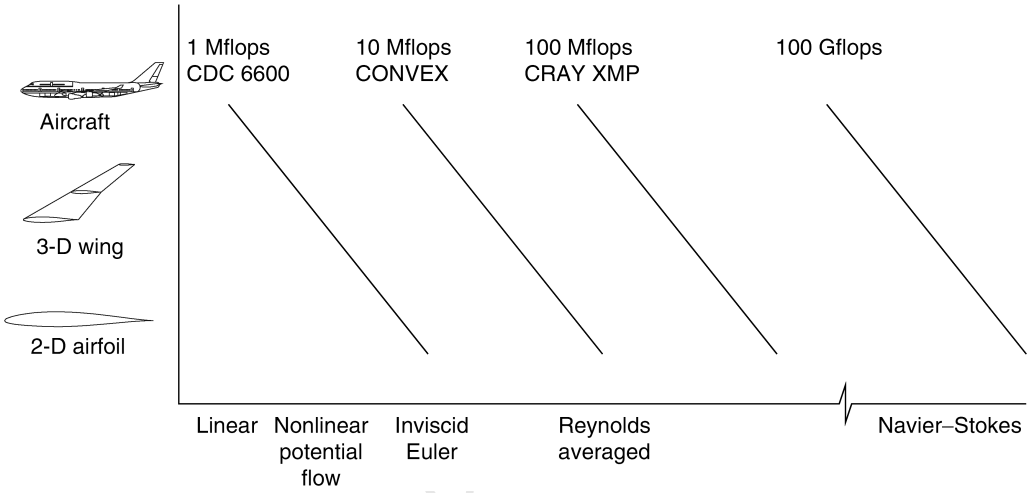


Figure 3.2: Complexity of the problems that can be treated with different classes of computer [21]

In this chapter some of the theories behind the aerodynamic models used in this dissertation will be described. Following a brief summary of the conservation equations that most fully describe fluid flow in §3.2, §3.3 and §3.4 describe the most basic linear theory and some extensions. Finally §3.5 to §3.7 cover some advanced topics.

3.2 Origin of Fluid Flow Equations

Summarised from §3.2 of [19]

Using standard tensor notation each conservation law can be cast into the form

$$\frac{\partial w}{\partial t} + \frac{\partial f_i}{\partial x_j} - \frac{\partial f_{v_i}}{\partial x_j} = 0$$

where f_i are the inviscid (convective) fluxes and f_{v_i} are the viscous fluxes. For the mass equation,

$$w = \rho, \quad f_i = \rho u_j$$

For the ith momentum equation,

$$w = \rho u_j, \quad f_{ij} = \rho u_i u_j + p\delta_{ij}, \quad f_{v_i} = \sigma_{ij}$$

where σ_{ij} is the viscous stress tensor. For the energy equation

$$w = \rho E, \quad f_i = pHu_j, \quad f_{v_i} = \sigma_{jk}u_k = k\frac{\partial T}{\partial x_j}.$$

The pressure, density, and energy are related by

$$p = (\gamma - 1)\rho(E - \frac{1}{2}u_i u_i)$$

and

$$E = c_v T + \frac{1}{2}u_i u_i.$$

In the Navier-Stokes equations the viscous stresses are assumed to be linearly proportional to the rate of strain, so that

$$\sigma_{ij} = \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) + \lambda\delta_{ij}\left(\frac{\partial u_k}{\partial x_k}\right)$$

where μ and λ are the coefficients of viscosity and bulk viscosity respectively. Also, the coefficient of thermal conductivity and temperature are governed by the following relations.

$$k = \frac{c_p\mu}{\mathrm{Pr}}, \quad T = \frac{p}{R\rho}$$

3.3 Potential Flow

Theories based on potential flow assume that the flow is everywhere irrotational. That is

$$\omega = \nabla \times \boldsymbol{v} = 0$$

This allows the velocity to be written as the gradient of a scalar potential,

$$\boldsymbol{v} = \nabla\phi \tag{3.1}$$

3.3.1 Compressible Potential Flow

Using equation 3.1 the most general form of the full potential equation can be cast in integral or differential form. The integral form is given by

$$\frac{\partial}{\partial t}\iiint_\Omega \rho \mathrm{d}\Omega + \iint_S \boldsymbol{n}\cdot\rho\nabla\phi \mathrm{d}S = 0$$

and the use of Gauss' divergence theorem gives the differential form,

$$\frac{\partial\rho}{\partial t} + \nabla\cdot\rho\nabla\phi = 0 \tag{3.2}$$

A relation that expresses the fluid density, ρ, as a function of the velocity components ϕ_x, ϕ_y and ϕ_z is needed to complete the formulation. An example can be derived from the definition of the speed of sound and the isentropic pressure/density relation according to Holst (equation (8) in [22]) as

$$\rho = \left[1 + \frac{\gamma - 1}{2}\left(M_\infty^2 - 2\phi_t - \phi_x^2 - \phi_y^2 - \phi_z^2\right)\right]^{\frac{1}{\gamma - 1}}$$

Several other forms of this relation exist depending on the nondimensionalisation used. For example, the steady flow version of the full potential equation in which density is nondimensionalised with the stagnation density, ρ_{stag}, and velocity components are nondimensionalised with the critical speed of sound a^* is given by

$$(\rho\phi_x)_x + (\rho\phi_y)_y + (\rho\phi_z)_z = 0$$

The density-velocity relation is then given by

$$\rho = \left[1 - \frac{\gamma - 1}{\gamma + 1}\left(\phi_x^2 + \phi_y^2 + \phi_z^2\right)\right]^{\frac{1}{\gamma - 1}}$$

3.3.2 Steady Incompressible Potential Flow

When the flow is incompressible,

$$\nabla \cdot \boldsymbol{v} = 0$$

so that

$$\nabla^2 \phi = 0 \tag{3.3}$$

since $\nabla \times (\nabla\phi) = 0$ for ϕ a scalar function.

Equation 3.3 is Laplace's equation, a second-order linear partial differential equation. Due to its linearity, ϕ can be composed of the sum of any number of solutions. In this way the principle of superposition is used to model complex flows from one or more types of elementary flows. This forms the basis behind several analytical approaches, for example thin wing theory, panel methods, and lifting line theory. Some elementary flows are shown in Fig. 3.3.

3.3.3 Compressibility Corrections for Incompressible Flow

The perfect gas equation, $p = \rho RT$, gives that for a change in pressure there will generally be a change in density and temperature. If a gas flows over a body resulting in large changes in pressure, the density will change significantly and the assumptions of incompressible flow become invalid. For speeds of around Mach 0.3 and below, assumptions of incompressibility do not add significant error. For speeds above this and up to the *critical Mach number*, M_{crit}, a compressibility

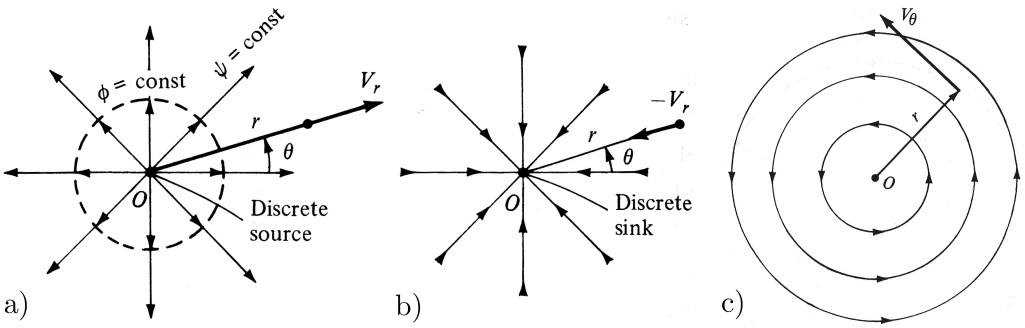


Figure 3.3: Three elementary flows a) source, b) sink, and c) vortex [2]

factor can be introduced. Beyond M_{crit} the flow becomes transonic and compressibility factors become invalid.

There are a number compressibility factors, the first of which was formulated by Prandtl and Glauert, given by

$$C_p = \frac{C_{p_0}}{\sqrt{1 - M_\infty^2}}$$

This was later refined by von Kármán and Tsien to

$$C_p = \frac{C_{p_0}}{\sqrt{1 - M_\infty^2} + \left[\frac{M_\infty^2}{1+\sqrt{1-M_\infty^2}}\right] \frac{C_{p_0}}{2}} \tag{3.4}$$

and again more recently by Laitone to

$$C_p = \frac{C_{p_0}}{\sqrt{1 - M_\infty^2} + M_\infty^2 \left(1 + \frac{\gamma-1}{2} M_\infty^2\right) \sqrt{1 - M_\infty^2} \frac{C_{p_0}}{2}}$$

Figure 3.4 shows a comparison between the corrections described by the above equations. These computations were performed by the author and clearly indicate significantly different results predicted by each method.

3.3.4 Transonic Small Disturbance Theory

Summarised from [22]

Assuming small perturbations in transonic flow, equation 3.2 can be closed using the integrated unsteady Bernoulli equation, the speed of sound definition, and the isentropic density-pressure relation, which results in the transonic small disturbance (TSD) equation. Potential formulations are irrotational and isentropic, which remains a good assumption for shocked flows provided the shock waves are weak. Here "weak" means that the normal-shock Mach number component, M_n, is less than around 1.3. For airliner cruise conditions the existence of strong shock waves is very undesirable, so this restriction on M_n should not limit the usefulness of these equations here. Fig. 3.5 shows generally allowable situations for full po-

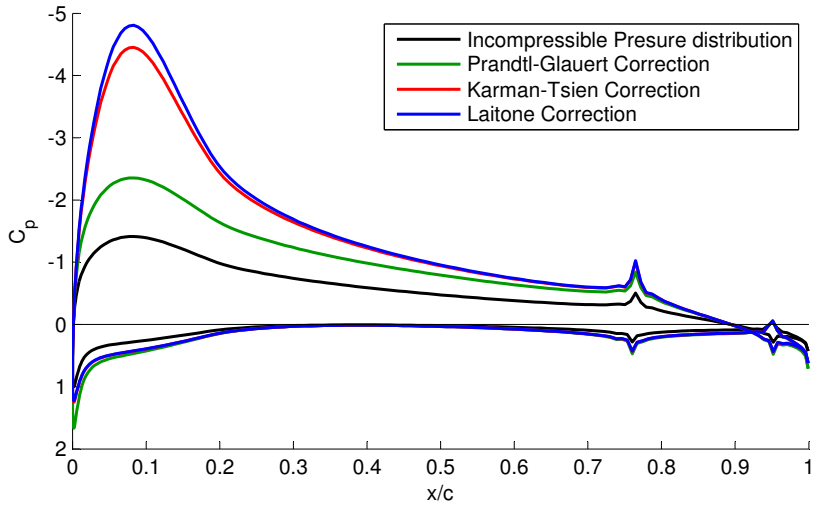


Figure 3.4: Comparison of compressibility corrections for flow over a double-flapped NACA 23012 airfoil at $M_\infty = 0.8$ with $\alpha = 4°$, $\delta_a = 4°$ and $\delta_t = -4°$.

tential simulations. In order to obtain physically realistic results analyses should also be confined to shapes and flows in which viscous effects (like flow separation) are minimal.

The TSD equation appears below. (See §2.5 of [22] for a derivation.)

$$\left[\left(1 - M_\infty^2\right)\varphi_x - M_\infty^2\left(\gamma + 1\right)\frac{\varphi_x^2}{2u_\infty}\right]_x + \varphi_{yy} = 0 \tag{3.5}$$

where φ is the small-disturbance velocity potential, defined by

$$\nabla\varphi = \boldsymbol{v} - \boldsymbol{v}_\infty$$

The nonlinearity of these equations means that solution by a direct method is impossible, and iterative numerical solution schemes must be used.

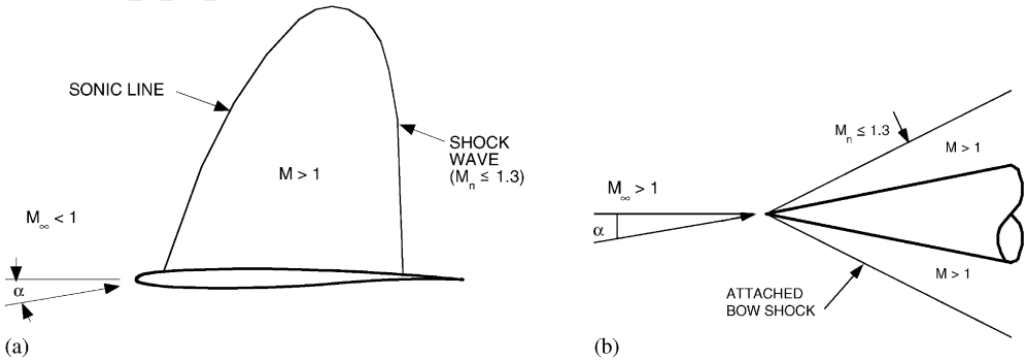


Figure 3.5: Allowable flow regimes for application of the full potential equations, showing a) transonic and b) hypersonic flow fields. [22]

3.3.5 The Kutta Condition

The application of potential flow theory to the solution of airflow around an airfoil posed a significant problem. Figure 3.6 shows the unmodified solution to Laplace's equation. Here no circulation is generated and the airfoil provides no lift. Examination of actual flow patterns around an airfoil showed a different picture, where the airflow leaves the trailing edge smoothly, and does not flow back over the top of the airfoil as seen in Fig. 3.6. The behaviour of real flows exhibits a pattern more like that seen in Fig. 3.7. This pattern can be achieved theoretically by the application of the Kutta condition, which states for incompressible flow that at the trailing edge $V_1 = V_2$, for V_1 the velocity parallel to the upper surface, and V_2 the velocity parallel to the lower surface. This yields, for the value of circulation at the trailing edge

$$\gamma(TE) = 0$$

The above equation is the Kutta condition, expressed in terms of the strength of a vortex distribution.

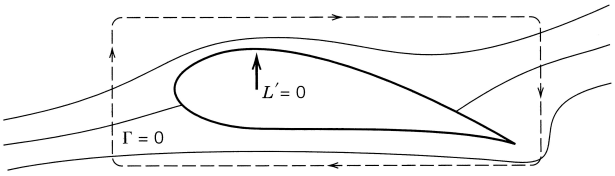


Figure 3.6: An airfoil in an inviscid flow with no condition of flow over the trailing edge [18]

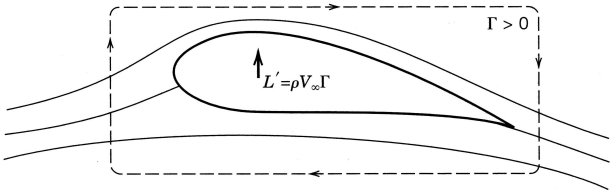


Figure 3.7: Application of the Kutta condition, producing circulation around the airfoil [18]

3.4 Boundary Layer Theory

Summarised from [23] unless otherwise stated

The potential flow equations presented in the preceding sections suffer from further discrepancies with reality, this time in the form of D'Alembert's paradox — that is, they predict zero drag. Although this contradiction was realised as early as 1752, the problem remained unsolved for practical aerodynamics until Ludwig Prandtl presented his boundary layer theory in 1904 [23]. Prandtl showed that flow could be separated into two regions – a thin, viscous, so-called boundary

layer, adjacent to the body, and the remaining flow where the inviscid solution remains a feasible approximation.

The no-slip condition for viscous flows states that the velocity at the wall must be zero, leading to a velocity profile like that shown in Fig. 3.8. Due to this velocity gradient normal to the wall even a fluid with a low viscosity can induce non-negligible shear stresses, since $\tau = \mu \frac{\partial u}{\partial y}$. This means that viscosity must be accounted for in order to reproduce accurate representations of near-wall flows.

Boundary-layer theory presented the first means to theoretically introduce viscosity to aerodynamic analyses, thus providing a way to calculate drag.

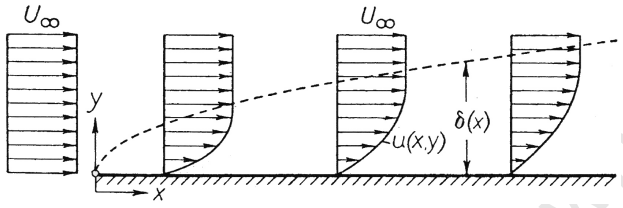


Figure 3.8: Laminar boundary layer along a flat plate at zero incidence showing the velocity profile which develops as a result of the no-slip condition. (The scale of the transverse axis has been greatly exaggerated.) [23]

3.4.1 Boundary Layer Development

Figure 3.9 shows a typical boundary layer for a simple airfoil. Flow in the boundary layer is initially laminar, but transitions to turbulent flow at some point depending on several factors. These include the pressure gradient, Reynolds number, freestream turbulence level, and surface roughness. The physical process of transition is dealt with more fully in §3.5. Separation can also occur in areas of sufficiently adverse pressure gradients, where the boundary layer thickens substantially and a loss of lift is encountered. Although large-scale separation occurs at high angles of attack and is responsible for the onset of stall, separation can occur at sharp edges or sudden airfoil profile gradient changes. Separation is discussed in greater detail in §3.6.

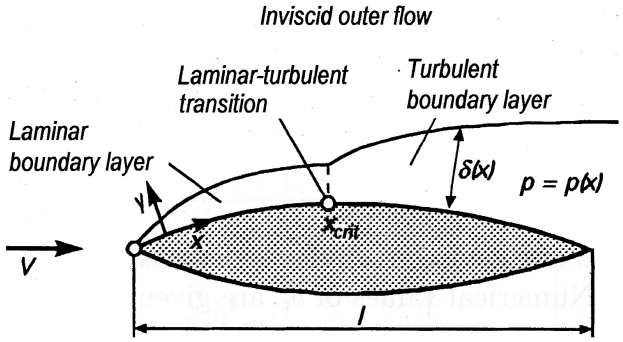


Figure 3.9: Typical boundary layer development over a subsonic airfoil at small angles of attack [23]

3.4.2 Pressure Distribution In The Boundary Layer

An assumption of boundary layer theory is that the pressure given by the inviscid solution for the outer flow can be applied throughout the boundary layer. Any variation in pressure normal to the airfoil surface could arise only due to centrifugal forces from streamline curvature. However, the radius of curvature of the airfoil is much greater than the boundary layer thickness for large Reynolds numbers, so these forces can be considered to be negligible.

This pressure distribution can have an important impact on the characteristics of the boundary layer. The pressure distribution will affect the point of transition from laminar to turbulent flow, as well as the onset of any separation that may occur.

3.4.3 Integral Method Boundary-Layer Equations

Summarised from [23] unless otherwise stated

The most common means of predicting boundary-layer flows is using a certain family of boundary layer analyses called integral methods, where momentum and continuity equations are integrated across the boundary layer. The governing boundary-layer equation is cast in terms of four variables which are described here before the final boundary-layer equation is presented.

Displacement thickness, δ^*

The boundary layer acts to displace the outer flow slightly away from the body. The reason for this can be seen in Fig. 3.10, where the conservation of mass must be satisfied even with the retardation of the flow in the boundary layer. Since no flow crosses the outer streamline its distance from the body must increase so that the mass flow rate is maintained in the boundary layer. This distance is given by δ^* and is called the displacement thickness. We have for a unit width that

$$\int_0^h \rho u_\infty dy = \int_0^\delta \rho u dy \tag{3.6}$$

where δ is the boundary layer thickness, and $\delta = h + \delta^*$, as shown in Fig. 3.10. Here h is a datum for the height of the equivalent freestream volume flow as that within the boundary layer.

Integration and algebraic manipulation of equation 3.6 yields the displacement thickness, where

$$\delta^* = \int_0^\delta \left(1 - \frac{u}{u_e}\right) dy$$

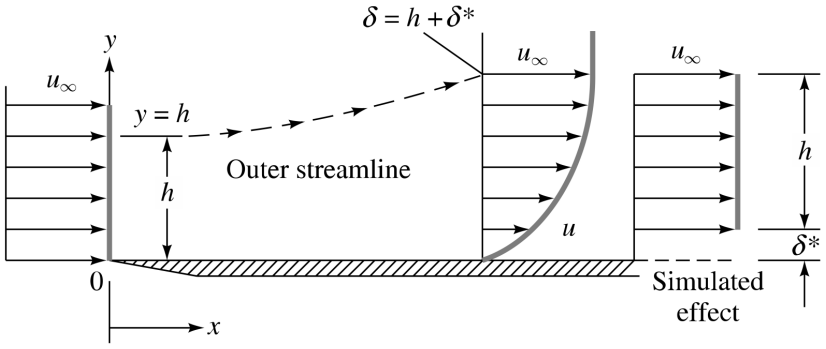


Figure 3.10: A decrease of velocity in the boundary layer displaces the outer flow away from the body [24]

Momentum thickness, θ

The momentum thickness gives a measure of the drag force produced by viscous effects in the boundary layer. Physically it can be thought of that $\delta^* + \theta$ is the thickness of a layer of the outer flow that has a momentum flow rate equal to the reduction in flow rate caused by the boundary layer. This is shown schematically in Fig. 3.11.

The momentum thickness is given by

$$\theta = \int_0^{\delta} \frac{u}{u_e}\left(1 - \frac{u}{u_e}\right) dy$$

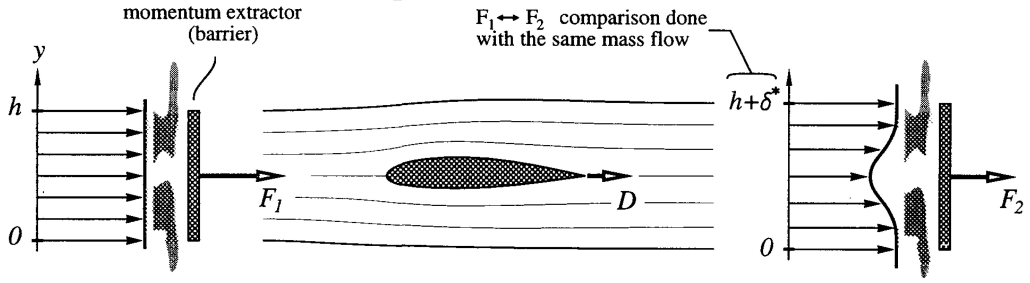


Figure 3.11: Physical interpretation of momentum thickness, θ, where $F_1 = D + F_2$ [25]

Shape Factor, H

The shape factor is simply the ratio of displacement thickness to momentum thickness, i.e.

$$H = \frac{\delta^*}{\theta}$$

For laminar flows $2 < H < 3$, and for turbulent flows $1.5 < H < 2$. Values of $H > 3$ indicate separation, with the separation point being at $H \approx 3.7$.

Skin Friction Coefficient, c_f

The skin friction coefficient is calculated directly from the shear stress on the body surface, where

$$c_f = \frac{\tau_w(x)}{\frac{1}{2}\rho u_e^2}$$

These are given, for a plat plate at zero angle of incidence, for laminar flow by

$$c_f = \frac{0.664}{\sqrt{\mathrm{Re}}}\sqrt{\frac{l}{x}}$$

and for turbulent flow by

$$c_f = 2\left[\frac{\kappa}{\ln \mathrm{Re}_x} G(\ln \mathrm{Re}_x)\right]^2$$

where the Karman constant, $\kappa = 0.41$.

This gives an estimate of the skin friction drag over an airfoil surface.

Momentum Integral Equation

We can now express the momentum integral equation for two-dimensional incompressible laminar or turbulent boundary layer flows in terms of the previously defined quantities. The result is as follows: [23]

$$\frac{d\theta}{dx} + (H+2)\frac{\theta}{u_e}\frac{du_e}{dx} = \frac{c_f}{2} \tag{3.7}$$

Equation 3.7 has too many unknows to be useful on its own, and must be supplemented by other equations. These methods involve assumptions about the form of the velocity profile and/or data fitting [26]. The following sections (§3.4.4 and §3.4.5) describe methods used by the software utilised later in this dissertation to close equation 3.7.

3.4.4 Thwaites' Method for Laminar Boundary Layers

Thwaites' method is an empirical formulation based on the observation that laminar boundary layers obey the following relationship

$$\frac{u_e}{\nu}\frac{d}{dx}\left(\theta^2\right) = A - B\frac{\theta^2}{\nu}\frac{du_e}{dx}$$

with $A = 0.45$ and $B = 6$. This equation is analytically integrated to yield θ when u_e is known. H is given by

$$H = \begin{cases} 2.61 - 3.75\lambda + 5.24\lambda^2 & \text{for } 0 \geq \lambda \leq 0.1, \\ 2.472 + \frac{0.0147}{0.107+\lambda} & \text{for } -0.1\lambda \leq 0. \end{cases}$$

where

$$\lambda = \frac{\theta^2}{\nu}\frac{du_e}{dx}$$

The wall shear stress is given by

$$\tau_w = \frac{\mu u_e}{\theta}\left(\lambda + 0.09\right)^{0.62}$$

3.4.5 Entrainment Methods for Turbulent Boundary Layers

"Entrainment" is the term used to describe the method by which turbulence is spread from a turbulent region of flow into an adjacent non-turbulent and substantially irrotational region [27]. Here the non-turbulent flow is said to undergo a process of entrainment due to the turbulent flow.

Entrainment methods have both theoretical and empirical components and are used to numerically predict growth of a turbulent boundary layer. They are particularly useful when integral boundary-layer methods are applied to an otherwise inviscid flow, and the two panel-method based programs used in this project use entrainment methods. These two methods used will be discussed here.

Head's Entrainment Method

Summarised from [27]

Head's method was put forward in the late 50s and used extensively by the Royal Aircraft Establishment (RAE) in the UK [27, 28]. It defines a new form parameter, $H_{\delta-\delta^*}$, based on the boundary layer thickness, displacement thickness, and the momentum thickness, where

$$H_{\delta-\delta^*} = \frac{\delta - \delta^*}{\theta}$$

This parameter is assumed to evolve as

$$\frac{1}{u_e}\frac{d}{dx}\left[u_e\left(\delta - \delta^*\right)\right] = f(H_{\delta-\delta^*})$$

The parameters commonly used are from Cebeci and Bradshaw [29], who found by experiment that

$$f(H_{\delta-\delta^*}) = 0.0306\left(H_{\delta-\delta^*} - 3.0\right)^{-0.6169}$$

and

$$H_{\delta-\delta^*} = \begin{cases} 0.8234\left(H - 1.1\right)^{-1.287} + 3.3, & \text{for } H \leq 1.6 \\ 1.5501\left(H - 0.6778\right)^{-3.064} + 3.3, & \text{for } H > 1.6 \end{cases}$$

Green's Entrainment Method

Summarised from [28]

Green's method was developed in 1977 as a consequence of the emergence of more accurate entrainment methods. Here an additional ODE is introduced so that turbulence stresses were treated independently of the local mean velocity profile. The third equation introduced by Green was derived from a turbulent kinetic energy equation which defines the peak shear stress, and which was converted into differential form to yield a rate equation for the entrainment coefficient. This equation is known as the "lag" equation, with the method being a "lag-entrainment" method.

The three equations used are the von Kármán integral boundary-layer equation, given by equation 3.7, the entrainment equation,

$$\theta \frac{dH}{dx} = \frac{dH}{dH_{\delta-\delta^*}} \left(C_E - H_{\delta-\delta^*} \frac{1}{u_e} \frac{d}{dx}(u_e \theta) \right)$$

and the lag equation,

$$(C_E)_{eq} = H_{\delta-\delta^*} \left[\frac{c_f}{2} - (H+1) \left(\frac{\theta}{u_e} \frac{du_e}{dx} \right)_{eq} \right]$$

where C_E is the entrainment coefficient.

3.4.6 Predicting Drag

A useful way to predict drag is to use the dissipation coefficient, C'_D. This is defined as

$$C'_D = \frac{1}{\rho_e u_e^3} \int \tau \frac{\partial u}{\partial y} \mathrm{d}y$$

where $\tau = \mu \frac{\partial u}{\partial y}$. One of the programs used later, namely XFoil, uses this implementation to calculate total drag.

This is formulated with

$$C'_D = \frac{C_f}{2} U_S + C_\tau \left(1 - U_S\right)$$

where the shear coefficient C_τ is a measure of the shear stresses in the wake layer, and U_S is an equivalent normalised wall slip velocity. C_f depends only on the local boundary layer parameters, whereas C_τ, representing the Reynolds stresses, is known to respond relatively slowly. In XFoil C_τ is expressed in a simplified form of the stress-transport equation of Bradshaw and Ferriss [30],

$$\frac{\delta}{C_\tau} \frac{\mathrm{d}C_\tau}{\mathrm{d}\xi} = 4.2 \left(C_{\tau_{eq}}^{\frac{1}{2}} - C_\tau^{\frac{1}{2}} \right)$$

3.5 Transition

The behaviour of flow past a body is strongly dependent on whether the boundary layer is laminar or turbulent. The transition from laminar to turbulent is most importantly affected by the Reynolds number, pressure distribution of the outer flow, surface roughness, and turbulence intensity of the outer flow. In flow past a slender body the pressure gradient has a considerable effect on the transition location, where the boundary layer generally remains laminar in regions where the pressure decreases (accelerated flow), but becomes turbulent in even the smallest adverse pressure gradients [23].

In this section the mechanisms of transition will be described, and the behaviour of transition for a general airliner wing will be scrutinised. One of the most commonly-used methods for predicting transition will also be examined.

3.5.1 Mechanisms of Transition

Tollmein-Schlichting waves

In the large majority of situations the onset of turbulence in a boundary layer can be described by a process that begins with two-dimensional Tollmein-Schlichting (T-S) waves and ultimately leads to fully turbulent flow. This is described by the sketch in Fig. 3.12, and is sometimes called "natural" transition for purposes of differentiation [31]. T-S waves are inherently two-dimensional, and can be described by linear theory. e^N methods were developed in the mid-1950s to describe T-S waves, and model incompressible two-dimensional flow, but have recently been developed to handle more complex flows [32].

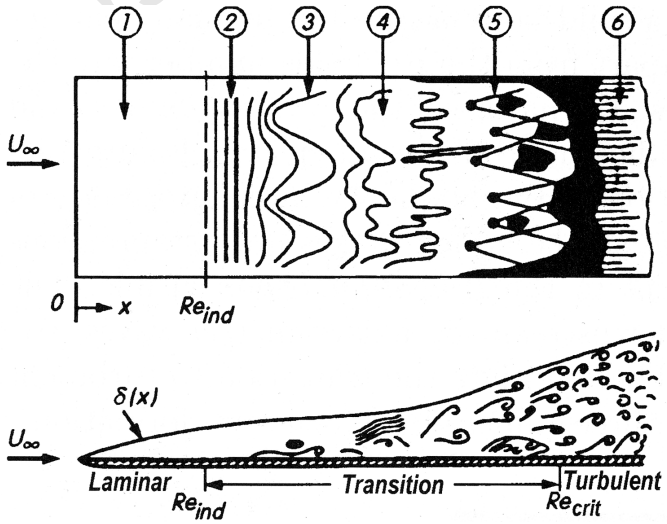


Figure 3.12: Laminar to turbulent transition in the boundary layer of a flat plate at zero incidence. (1) - stable laminar flow (2) - development of Tollmein-Schlichting waves (3) - vortex formation (Λ-structures) (4) - vortex decay (5) - formation of turbulent spots (6) - fully turbulent flow [23]

Attachment-line and crossflow instabilities

For swept wings there are two additional mechanisms that can cause transition. So-called attachment-line instabilities flow down the leading edge along the stagnation line, where perturbations which originate upstream, at the wing/fuselage juncture for example, grow and cause the flow to become turbulent. This can lead to turbulence spreading over the entire wing surface. The geometry here is illustrated in Fig. 3.13 which shows a wing with sweep angle Λ, normal velocity u_∞ and sweep velocity w_∞. The attachment line is oriented along the z-axis. [33]

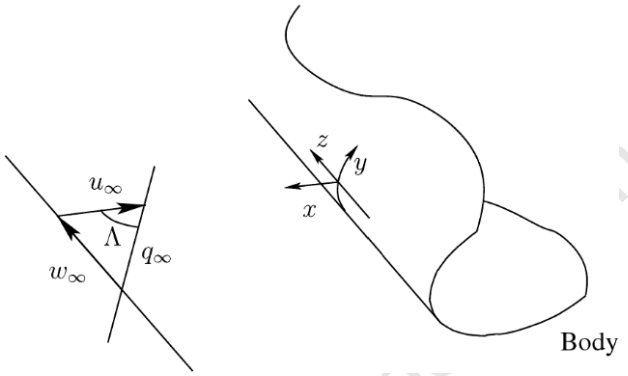


Figure 3.13: Attachment line for a swept wing, oriented along the z-axis [33]

Swept wings also suffer from crossflow instabilities that can lead to turbulence. A sketch of the formation of crossflow instabilities is shown in Fig. 3.14. Crossflow instabilities tend to dominate in regions of favourable pressure gradients, such as are found near a wing's leading edge [34].

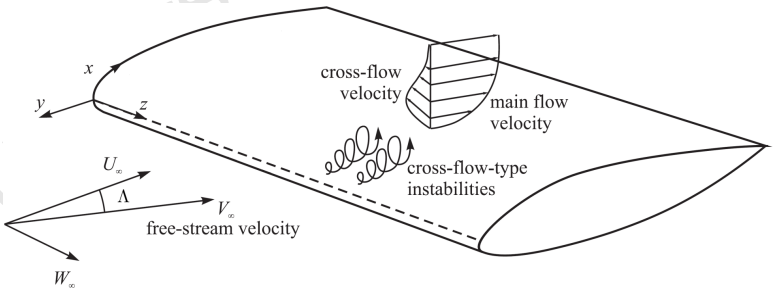


Figure 3.14: The formation of cross-flow instabilities due to sweep angle Λ [35]

Bypass transition

If there is a small (1% [36]) level of freestream turbulence intensity, excessive vibration, or wall roughness [37] bypass turbulence results without the appearance of Tollmein-Schlichting waves. These linear instabilities are bypassed and instead non-linear processes drive amplification of the disturbances [31].

3.5.2 Fully Turbulent Assumption for Transport Aircraft

Reference 38 states that "for transport aircraft with swept wings of mostly metal construction that don't use any laminar flow control to delay transition, the fully turbulent flow assumption is often considered to be a safe one." However, Fig. 3.15 shows an analysis using infrared thermography which indicates extended regions of laminar flow over the slat of a Boeing B737-100 at cruise conditions (see the caption for details). This indicates that the fully turbulent flow assumption should be applied with caution [38].

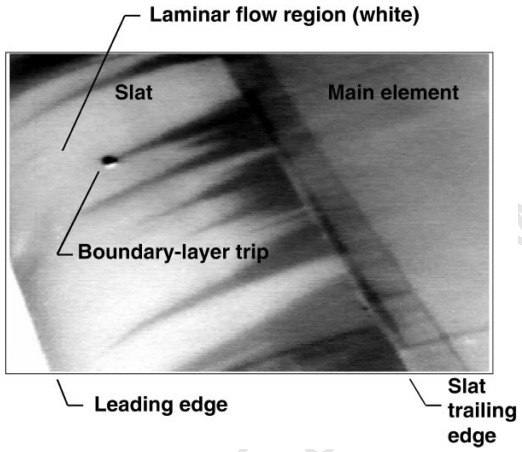


Figure 3.15: Transition on a Boeing 737 slat at cruise conditions ($\alpha = 3.1°$, $M = 0.68$, $h_p = 10\,900\ m$). The darker regions show turbulent flow, set off by a trip and insect contamination. [39]

3.5.3 Michel's Criterion For Predicting Turbulence

Summarised from [40]

A popular method for predicting transition for low Reynolds number flows is Michel's criterion. This empirical model correlates local values of momentum thickness with position of the transition point, so that transition will occur when

$$\mathrm{Re}_\theta \geq 1.174 \left(1 + \frac{22400}{\mathrm{Re}_x}\right) \mathrm{Re}_x^{0.46}$$

where Re_θ is the momentum thickness Reynolds number, and Re_x is the chord Reynolds number measured from the stagnation point. The relationship holds for $10^5 \leq \mathrm{Re}_x \leq 40 \times 10^6$.

3.5.4 e^N Methods For Predicting Turbulence

Summarised from [32]

So-called e^N methods were developed by Smith and Gamberoni [41], and van Ingen [42], first for low-speed 2D flows, and later extended to compressible 3D

flows. They are based on linear theory, so many fundamental aspects of transition are not accounted for.

The principle of the e^N method can be understood from Fig. 3.16. A wave which propagates downstream with fixed frequency f_1 will first pass through a stable region. It is damped up to x_0, amplified up to x_1, and damped again downstream of x_1. At any x the total amplification rate can be defined by

$$\ln\left(\frac{A}{A_0}\right) = \int_{x_0}^{x} -\alpha_i \mathrm{d}x$$

where A is the wave amplitude, the index 0 refers to the streamwise position where the wave becomes amplified, and α_i is a spatial growth rate parameter. The envelope of the total amplification curves is given by

$$N = \max\left[\ln\left(\frac{A}{A_0}\right)\right]$$

In a low-disturbance incompressible environment it is assumed that transition occurs when N reaches some critical value in the range 7-10. That is, when the locally most unstable frequency is amplified by a factor e^7 to e^{10}. This is called the envelope method.

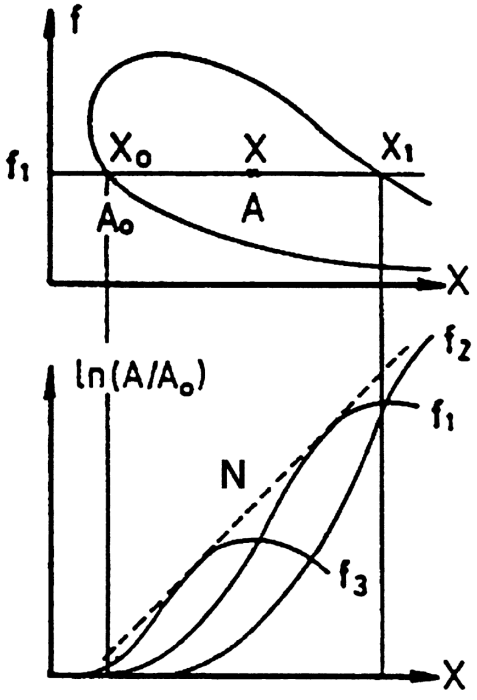


Figure 3.16: Typical stability diagram in physical coordinates defining the total growth rate and the envelope curve. [32]

3.6 Separation

Summarised from [23]

Separation is characterised by the occurrence of reversed flow, where an extreme thickening of the boundary layer takes place producing large regions of highly turbulent flow. This can cause a severe increase in drag and reduction of lift, depending on the extent of the separated region.

Separation can only take place in regions of adverse pressure gradient, where a point of inflection occurs in the velocity profile (see §7.6 of [24] for a rigorous proof). Figure 3.17 shows streamlines, the velocity profile, and boundary layer thickness for flow around a typical point of separation. At the critical adverse pressure gradient the velocity gradient perpendicular to the wall vanishes. That is,

$$\tau_w = \mu \left(\frac{\partial u}{\partial y} \right)_w = 0$$

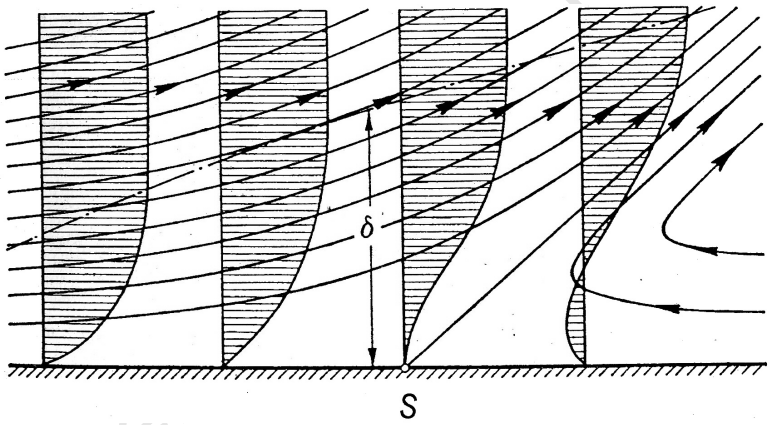


Figure 3.17: Streamlines within the boundary layer around the separation point, S. [23]

For airfoils, separation is responsible for the onset of stall, where a large separated region occurs at high angles of attack ($|\alpha| \gtrsim 15°$).

3.6.1 Laminar Separation Bubbles

Summarised from [43]

Laminar to turbulent transition can be set off by a separation bubble, which occurs predominantly in low Reynolds-number flows. These bubbles appear in the laminar boundary layer, the anatomy of which is shown in Fig. 3.18. Small disturbances present in the laminar flow are strongly amplified and rapid transition to turbulence takes place. The turbulence, in turn, creates a large momentum transport normal to the shear layer so that the flow reattaches to the surface.

Figure 3.19 show pressure distributions over two airfoils which show evidence of separation bubbles during low-speed flight. See caption for details.

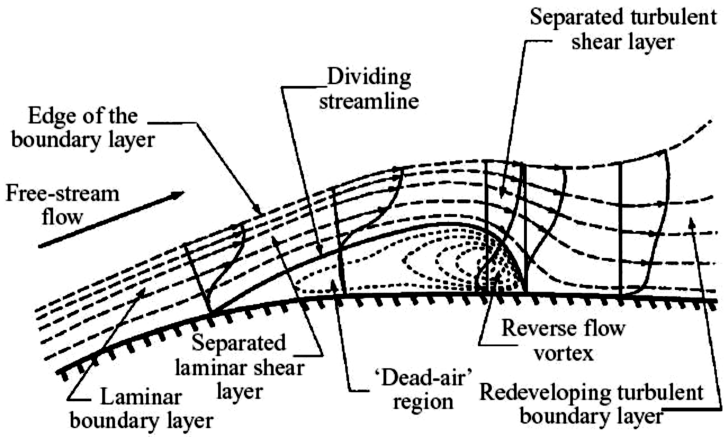


Figure 3.18: The anatomy of a laminar separation bubble [44]

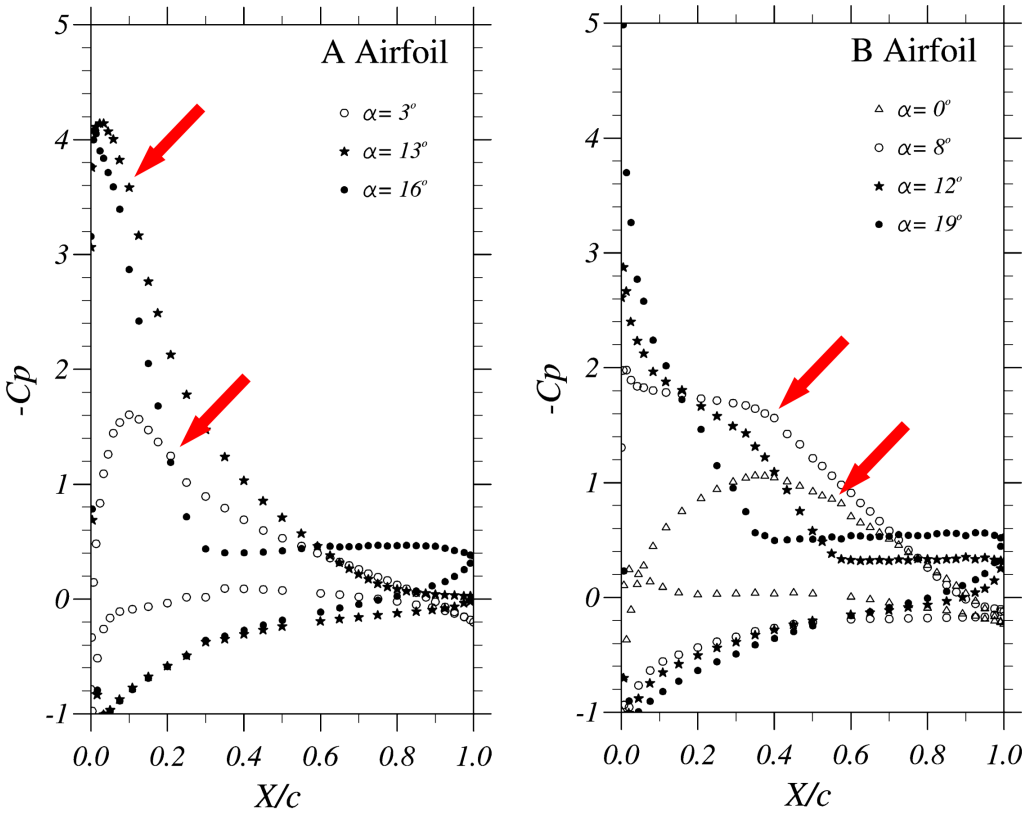


Figure 3.19: Separation bubbles measured during a collaborative study by AIRBUS and ONERA for the purposes of CFD validation. Measurements of two airfoils were recorded using F2 LDA pressure taps with $M_\infty = 0.15$ and $Re_c = 2.1 \times 10^6$. The red arrows point to evidence of transitional separation bubbles, the presence of which was also confirmed with oil flow visualisations. [45]

3.7 Turbulence Modelling

Complete solutions to the Navier-Stokes equations are excessively complex, and the problem is given by Bendiksen as due to "the abundance of excess dynamics" [46]. A large part of this is due to turbulence, and the complexity involved in

calculating the behaviour of eddies which exist in many length scales, all of which need to be resolved. Mesh sizes in space-time are quoted as needing to be of the order of Re^3, so that for a flow having $Re = 10^6$ about 10^{18} grid points are required, and is considered to be beyond the capabilities of any foreseeable computer [21].

Several approaches to this problem have been developed and this field is under continual development. Turbulence models can be categorised as follows [47]:

- RANS-based models
  - Linear eddy-viscosity models
  - Non-linear eddy viscosity models and algebraic stress models
  - Reynolds stress transport models
- Large eddy simulations
- Detached eddy simulations and other hybrid models
- Direct numerical simulations

A full discussion of the various kinds of turbulence models that exist, or their precise formulation, is beyond the scope of this dissertation. It should be noted, however, that no single turbulence model can be applied to all types of flow, with certain models being better suited to certain applications. The turbulence model discussed below was the one chosen for use in the CFD simulations undertaken.

3.7.1 The Spalart-Allmaras Turbulence Model

This section summarised from [48] §12.3.2 unless otherwise stated

The Spalart-Allmaras (S-A) turbulence model is a one-equation model that falls under "Linear eddy-viscosity models" in the RANS-based models category above. It was designed specifically for aerospace applications involving wall-bounded flows and has been shown to give good results for boundary layers subjected to adverse pressure gradients [48]. It was developed in 1992 by P.R. Spalart and S.R. Allmaras.

The model is described by the following equation, which in essence contains terms of diffusion, production and destruction.

$$\frac{\partial}{\partial t}\left(\rho\widetilde{\nu}\right)+\frac{\partial}{\partial x_i}\left(\rho\widetilde{\nu}u_i\right)=G_\nu+\frac{1}{\sigma_{\widetilde{\nu}}}\left[\frac{\partial}{\partial x_j}\left\{\left(\mu+\rho\widetilde{\nu}\right)\frac{\partial\widetilde{\nu}}{\partial x_j}\right\}+C_{b2}\rho\left(\frac{\partial\widetilde{\nu}}{\partial x_j}\right)^2\right]-Y_\nu+S_{\widetilde{\nu}}$$

where G_ν is the production of turbulent viscosity and Y_ν is the destruction of turbulent viscosity that occurs in the near-wall region due to wall blocking and viscous damping. $\sigma_{\widetilde{\nu}}$ and C_{b2} are constants and ν is the molecular kinematic viscosity. $S_{\widetilde{\nu}}$ is a user-defined source term.

The turbulent viscosity, μ_t, is computed from

$$\mu_t=\rho\widetilde{\nu}f_{v1}$$

where the viscous damping function, f_{v1}, is given by

$$f_{v1} = \frac{\chi^3}{\chi^3 + C_{v1}^3}$$

and

$$\chi \equiv \frac{\widetilde{\nu}}{\nu}$$

Turbulent Production

The production term, G_ν, is modelled as

$$G_\nu = C_{b1}\rho\widetilde{S}\widetilde{\nu}$$

where

$$\widetilde{S} = S + \frac{\widetilde{\nu}}{\kappa^2 d^2} f_{v2}$$

and

$$f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}}$$

Here C_{b1} and κ are constants, d is the distance from the wall, and S is a scalar measure of the deformation tensor. By default in Fluent, as in the original model proposed by Spalart and Allmaras, S is based on the magnitude of the vorticity:

$$S = \sqrt{2\Omega_{ij}\Omega_{ij}}$$

where Ω_{ij} is the mean rate-of-rotation tensor and is defined by

$$\Omega_{ij} \equiv \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i}\right)$$

Turbulent Destruction

The destruction term is modelled as

$$Y_\nu = C_{w1}\rho f_w \left(\frac{\widetilde{\nu}}{d}\right)^2$$

where

$$f_w = g\left[\frac{1 + C_{w3}^6}{g^6 + C_{w3}^6}\right]^{\frac{1}{6}}$$

$$g = r + C_{w2}\left(r^6 - r\right)$$

$$r \equiv \frac{\widetilde{\nu}}{\widetilde{S}\kappa^2 d^2}$$

C_{w1}, C_{w2}, and C_{w3} are constants, and $\widetilde{S}$ is given above.

Model Constants

The model constants C_{b1}, C_{b2}, $\sigma_{\widetilde{\nu}}$, C_{v1}, C_{w1}, C_{w2}, C_{w3}, and κ have the following default values:

$$C_{b1} = 0.1355, \quad C_b2 = 0.622, \quad \sigma_{\widetilde{\nu}} = \frac{2}{3}, \quad C_{v1} = 7.1$$

$$C_{w1} = \frac{C_b1}{\kappa^2}\frac{1 + C_{b2}}{\sigma_{\widetilde{\nu}}}, \quad C_{w2} = 0.3, \quad C_{w3} = 2.0, \quad \kappa = 0.4187$$

Wall Boundary Conditions

At walls, the modified turbulent kinematic viscosity, $\widetilde{\nu}$, is set to zero. When the mesh is fine enough to resolve the laminar sublayer, the wall shear stress is obtained from the laminar stress-strain relationship:

$$\frac{u}{u_\tau} = \frac{\rho u_\tau y}{\mu}$$

If the mesh is too coarse to resolve the laminar sublayer, it is assumed that the centroid of the wall-adjacent cell falls within the logarithmic region of the boundary layer, and the law-of-the-wall is employed:

$$\frac{u}{u_\tau} = \frac{1}{\kappa}\ln E\left(\frac{\rho u_\tau y}{\mu}\right)$$

where u is the velocity parallel to the wall, u_τ is the shear velocity, y is the distance from the wall, κ is the von Kármán constant (0.4187), and $E = 9.793$.

3.7.2 Mesh Size Requirement

The S-A model sets a requirement on the wall bounded grid spacing, where

$$y^+ = \frac{y}{\mu}\sqrt{\rho\tau_w}$$

Here y is the height of the cell normal to the wall. The value of y^+ dictates how wall shear stress is calculated. y^+ should either be of the order of 1, or $30 \approx< y^+ < 300$, with a preference for the latter option in the lower region of the bound.

4 Aerodynamic Software

4.1 Introduction

In this chapter a survey of aerodynamic software capable of predicting the aerodynamics of airfoils is undertaken, and several programs are chosen to be used. The aim is primarily to gain an understanding of different methods used, which aspects of the flow can be captured, and at what computational cost. In addition a suitable program is to be chosen which will be used in the optimisation procedure, covered later in this dissertation.

The programs which were chosen for use are described here to give an overview of their computational abilities as well as the kinds of outputs which they give. A more detailed comparison of their results and the flow characteristics they capture will appear in the following chapter.

4.2 Available Aerodynamic Software

An internet search was conducted to find open source or free software that solves airflow over an airfoil. Listed below are the programs found, the language in which they are programmed, the flow regimes they cover, whether they are two- or three-dimensional, and their basic method of solution.

AVL — Written in Fortran, AVL solves for subsonic flow over a 3D rigid body using a vortex lattic method for the lifting surfaces together with a slender-body model for fuselages and nacelles. For more information visit `http://web.mit.edu/drela/Public/web/avl/`

JavaFoil — Written in Java, JavaFoil solves for subsonic flow over a 2D airfoil using a panel method with linear varying vorticity distribution. The boundary layer parameters are solved with an integral method using the Eppler equations and criteria for transition and separation. For more information visit `http://www.mh-aerotools.de/airfoils/javafoil.htm`

PABLO — Written in Matlab, PABLO solves for subsonic flow over a 2D airfoil using a choice of a source, doublet, or vortex panel method, and includes a one-way boundary-layer analysis for drag prediction. For more information visit `http://www.nada.kth.se/~chris/pablo/pablo.html`

Tornado — Written in Matlab, Tornado solves for subsonic flow over a 3D airfoil using a vortex lattice method. For more information visit `http://www.redhammer.se/tornado/`

TSFoil — Written in Fortran, TSFoil solves the TSD equations over a 2D airfoil using a finite-difference method. For more information visit `http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19790004836_1979004836.pdf`

VLMpc — Written in Fortran, VLMpc solves for subsonic flow over a two-surface 3D airfoil using a vortex lattice method. For more information visit `http://www.aoe.vt.edu/~mason/Mason_f/VLMpc.exe`

XFoil — Written in Fortran, XFoil solves for subsonic flow with compressibility corrections over a 2D airfoil using a vortex element panel method with a coupled boundary layer. For more information visit `http://web.mit.edu/drela/Public/web/xfoil/`

4.3 Programs Chosen

In order to identify the level of complexity required to garner accurate results, as well as to gain a feel for what physical aspects each level of theoretical approach captures, three programs were more closely scrutinised. XFoil was chosen to represent linear potential flow due to its ubiquitousness and seemingly high regard. The optional inclusion of a strongly coupled boundary layer offers a useful comparison between inviscid and viscous flow. TSFoil offered the only available transonic analysis, and gives an insight into the physical aspects captured by non-linear inviscid potential flow.

It will be seen that none of the available software gives results that capture the full physics of the actual transonic conditions encountered in reality. However, in order to test the robustness of the optimisation procedure used later, PABLO was found useful in order to be integrated seamlessly with Matlab for the optimisation procedure. Although PABLO is not expected to yield a realistic optimum configuration, it allows for efficient data generation in order to evaluate the optimisation procedure used later, when 729 configurations must be analysed.

4.4 XFoil

XFoil was conceived as a design tool for the Daedalus Project, a successful endeavour by MIT to break the world record for human powered flight. Figure 4.1 shows the aircraft, which flew 115.11 kilometres in 3 hours and 54 minutes during its record-breaking flight [49]. Originally written by Mark Drela, it was completed in 1985, but since then it has received numerous revisions, up to v6.94 in 2001.

XFoil has been widely used, most notably for low Reynolds number analyses. It seems to be a firm favourite amongst radio-controlled modelers, and has recently gained favour for the analysis of wind [51–56] and water [57–60] turbine blades.

Figure 4.1: Daedalus human powered aircraft in flight. [50]

4.4.1 Analysis

This section summarised from [37]

XFoil uses a linear-vorticity stream-function panel-method to calculate the inviscid flow around an airfoil, strongly coupled with an integral-method boundary layer which predicts separation and drag. It also incorporates a Karman-Tsien compressibility correction, allowing good compressible predictions all the way to sonic conditions, but does not account for the effects of flow containing shock waves. [37] This combining of the inviscid solution and boundary layer equations yields a nonlinear elliptic system which is readily solved by a full-Newton method as described in reference 61.

Viscous Formulation

The boundary layer and wake are described with a two-equation lagged-dissipation integral-method formulation and an envelope e^N transition criterion. The boundary layer and wake are strongly interacted with the incompressible potential flow via a surface transpiration model, which permits proper calculation of limited separation regions. When an angle of attack is specified, the wake trajectory is taken from the inviscid solution in order to cut computational costs. Viscous effects will in general decrease lift and change the wake trajectory, but the effect of this approximation on overall accuracy is small. The manual states that for attached flows it is hardly perceptible, "and will be felt mainly near or past stall, where accuracy tends to degrade anyway" [37].

Figure 4.2 shows XFoil's graphical output window which displays C_p vs $\frac{x}{c}$ using the viscous analysis. The dashed line shows the inviscid solution, with the solid line giving the viscous solution. The manual states that difference between the two "is due to the modification of the effective airfoil shape by the boundary layers." This slightly larger effective shape is superimposed on the actual airfoil shape below the output plot. The gap between the two shapes is δ^*, which is around a third to half the total boundary-layer thickness [37]. The horizontal dotted line at $C_p \approx -1.3$ is $C_{p_{\text{crit}}}$, a measure of the associated sonic velocity, so any result which has a portion of the pressure distribution crossing this line indicates regions of supersonic flow, and should generally be regarded as invalid due to the inability of XFoil to model shocked flow.

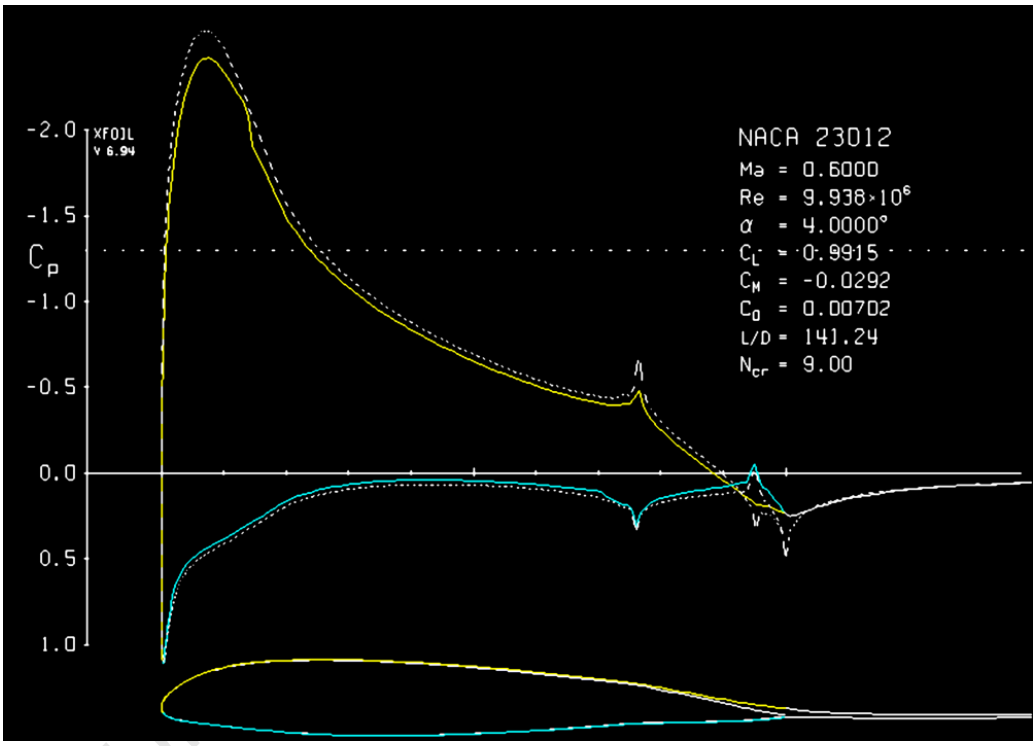


Figure 4.2: XFoil C_p vs $\frac{x}{c}$ graphical output for a viscous analysis of a NACA 23012 airfoil with $\delta_a = 4°$, $\delta_t = -4°$, $\alpha = 4°$ and $M = 0.6$. Control surfaces hinges are at $0.76c$ and $0.95c$ and fall on the mean camber line.

Transition

XFoil uses an e^N method to evaluate the onset of transition, which has the user-specified parameter N_{crit}. Typical values of N_{crit} are given in [37] and appear in Table 4.1. Transition can also be forced at an $\frac{x}{c}$ value, but the e^N method is always in effect so that transition is properly detected even if it occurs before the forced position. However, if an attempt is made to force transition before the stagnation point (for example, by setting $\frac{x_{\text{crit}}}{c} = 0$ on the lower surface for $\alpha > 0$), transition will occur naturally, and not be forced from the stagnation point. Setting an N_{crit} value of around 1 or lower will result in bypass transition.

Table 4.1: Typical N_{crit} values for XFoil's e^N transition calculation

Situation	$\boldsymbol{N_{\text{crit}}}$
sailplane	12-14
motorglider	11-13
clean wind tunnel	10-12
average wind tunnel	9
dirty wind tunnel	4-8

Calculation of Drag

Xfoil's viscous formulation calculates drag by the following equation:

$$C_D = \int 2\, C'_D \, \mathrm{d}s$$

where C'_D is the dissipation coefficient, and the integration is performed over both boundary layers and the wake. The XFoil User Manual, [37], justifies this calculation with the following:

> *$C'_D(x)$ is proportional to the local energy dissipation rate due to viscous shear and turbulent mixing. Hence, it indicates where on the airfoil drag is being created. It is in fact a much better indicator of drag production than $C_f(x)$, since C_f does not account for pressure drag. C'_D, on the other hand, accounts for everything.*

4.4.2 Running XFoil

XFoil can be run in two ways. When running the `exe` file in a Windows environment, a DOS window opens and all input is performed via a command prompt interface using an extensive menu system. This is described in detail in [37], with an example of the opening screen shown in Fig. 4.3. XFoil can also be run by an appropriate external program that utilises shell commands. This can be done in Matlab with the "bang" character (`!`) by inputting a text file containing what would usually be entered into the command prompt. An example of this process can be found in Appendix A.

XFoil has a built-in algorithm for creating NACA profiles, and includes a geometry design menu where any number of flaps can be created. The number of panels used in the computation can be adjusted to a maximum of 280, and are distributed in such a manner that the nodes are more concentrated around regions of greater curvature and near the leading and trailing edges. This can be seen in Fig. 4.4, where a greater node density is found over the leading and trailing edges as well as near the flap hinges.

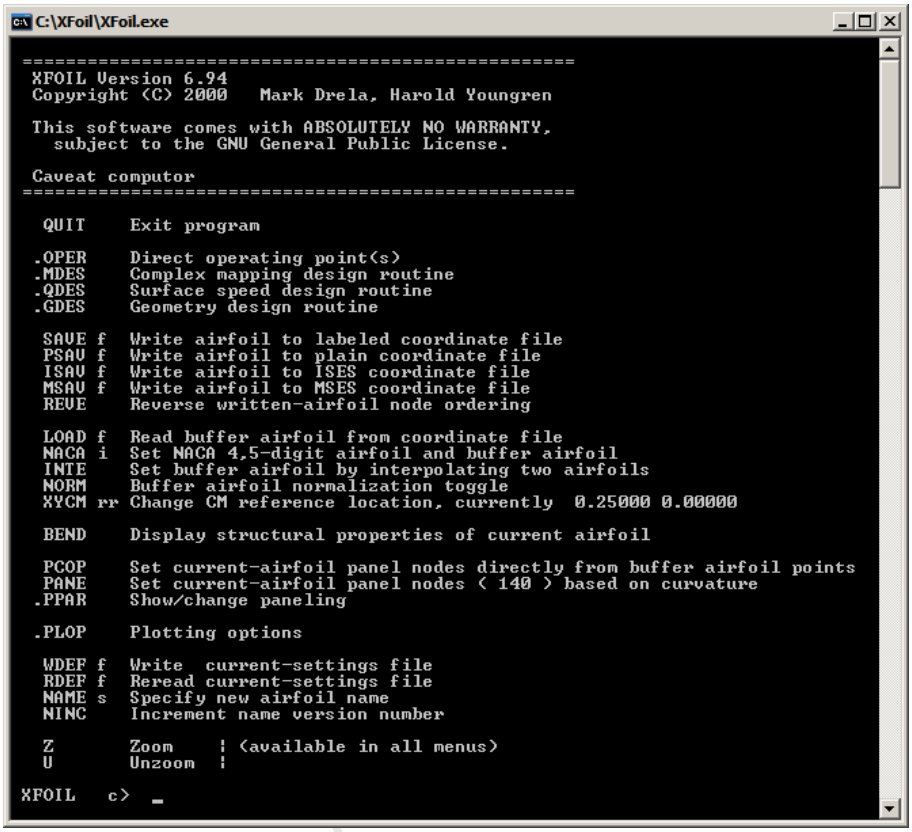


Figure 4.3: XFoil command prompt at program start

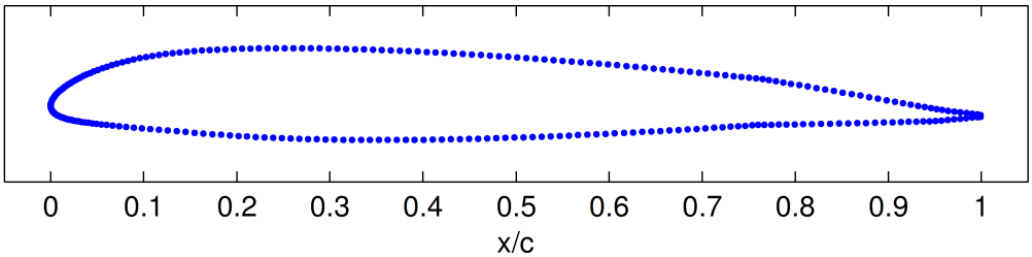


Figure 4.4: XFoil airfoil profile, output to a text file and plotted in Matlab. NACA 23012 airfoil with $\delta_a = 4°$ and $\delta_t = -4°$.

Graphs which are usually output to the screen can also be printed to a file in postscript (`.ps`) format, or saved as a table in text format. An example of the file output of the on-screen display seen in Fig. 4.2 can be seen in Fig. 4.5. Boundary layer parameters can also be examined on screen or output in this way.

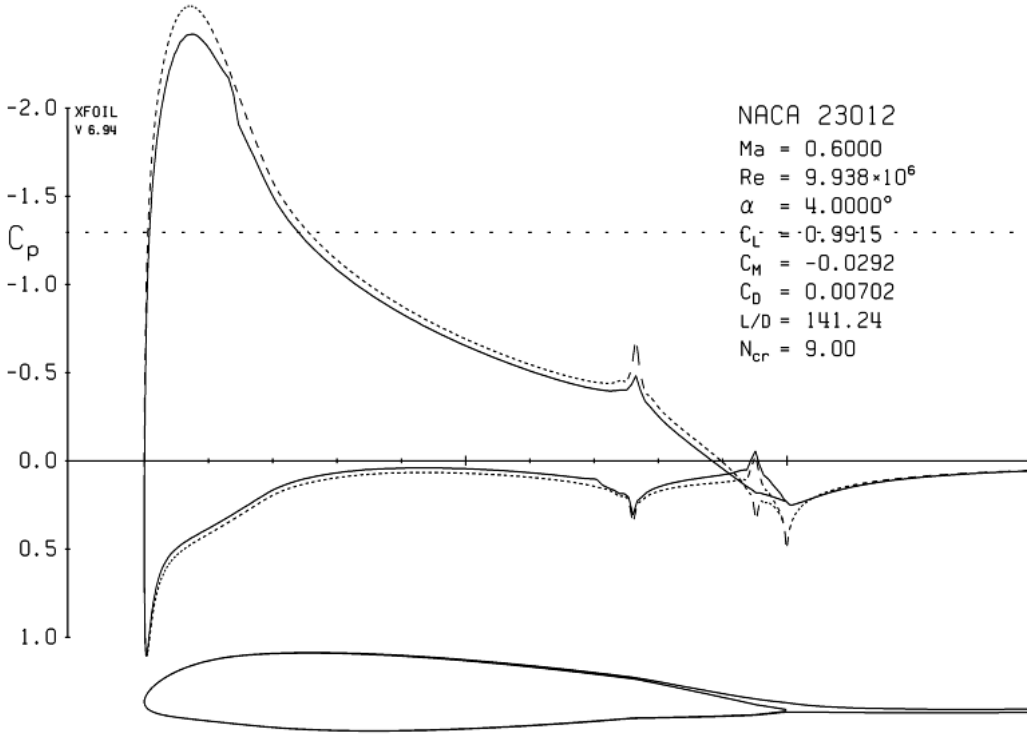


Figure 4.5: XFoil `.ps` output of result shown in Fig. 4.2

4.5 TSFoil

TSFoil was developed by Earll Murman, Frank Bailey, and Margaret Johnson at NASA Ames Research Center in the mid 1970s. It solves two-dimensional inviscid transonic flow for thin wings using the small-disturbance potential equations.

4.5.1 Analysis

The TSD equation (see equation 3.5 in §3.3.4) is solved using relaxation methods on three grids of different sizes, which are used in succession (coarse → medium → fine) to facilitate convergence. The numerical procedures are of the shock-capturing variety, so that the shock is spread over several mesh intervals instead of being an exact discontinuity. But use of the fully conservative difference equation ensures the correct shock jump is computed as the mesh is refined. [62]

Finite difference approximations are used to calculate the partial derivatives in equation 3.5. For subsonic flow a central difference scheme is used to calculate ϕ_{xx}, and for supersonic flow an upwind difference scheme is used. Since this mimics the actual physics of the flowfield, shocks are allowed to emerge naturally, hence the term "shock capturing" which is used to describe this method. [63]

Calculating Drag

Inviscid wave drag is predicted using a momentum integral method, and is given by

$$\frac{C_D}{2\delta^{\frac{5}{3}} M_\infty^{-n}} = \oint_C \left[\left(\kappa \frac{u^2}{2} - \frac{v^2}{2} - \frac{\gamma+1}{3} u^3 \right) \mathrm{d}\widetilde{y} - (uv)\mathrm{d}x \right] - \frac{\gamma+1}{12} \int_{S \text{ in } C} [u]^3 \mathrm{d}\widetilde{y}$$

using the control surfaces given by Fig. 4.6. Above, $\widetilde{y}$ is the transonically scaled y coordinate, n depends on the transonic scaling used (see Table 4-1 in [64]), and $[u]$ is the jump in ϕ_x across the shock waves S contained within C. [64]

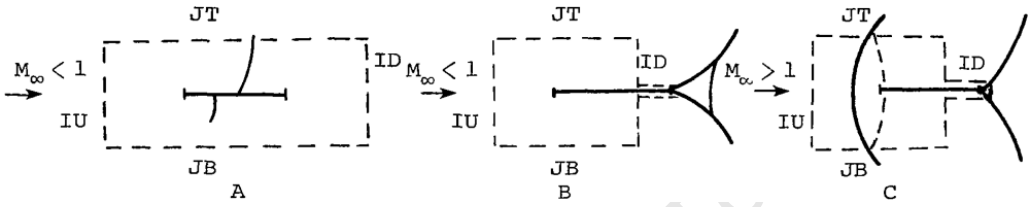


Figure 4.6: Control volumes for TSFoil inviscid drag calculations. [64]

4.5.2 Running TSFoil

The program runs using a command prompt interface, with the initial screen shown in Fig. 4.7.

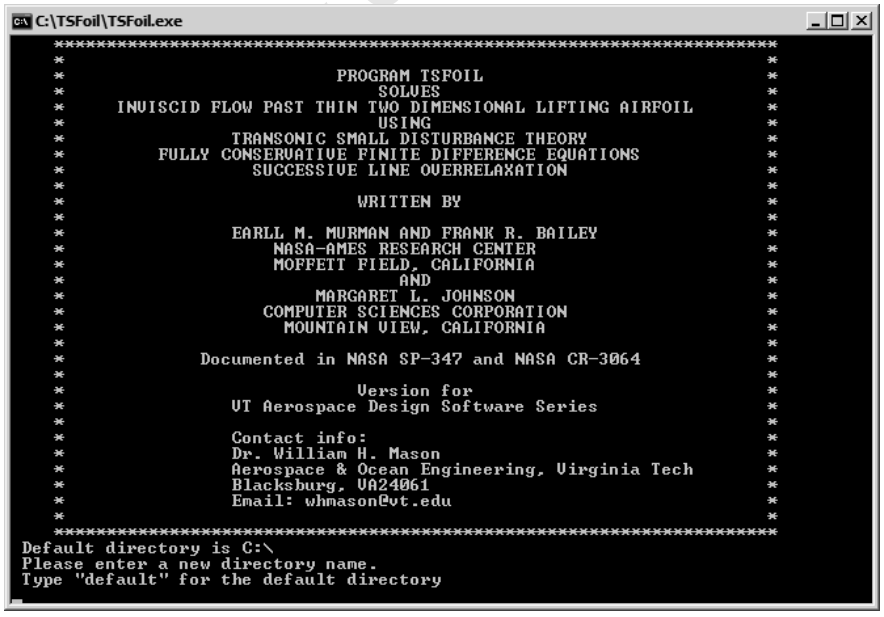


Figure 4.7: Opening screen of TSFoil

The only prompts are for a directory, an output file name, and an input file name. The TSFoil manual ([64]) contains a large variety of options than can be set within the input file. Here we simply present an example input file which was used to test the program (see Appendix B.1). Two output files are generated – one

which includes a thorough set of results and bears the user-defined file name (see Appendix B.2), and one containing the Mach number and C_p distribution against $\frac{x}{c}$ for each of the upper and lower surfaces, simply named `fort.1` (see Appendix B.3).

Preliminary tests were done on a NACA 23012 airfoil with $\delta_a = 4°$ and $\delta_t = -4°$. The profile was generated in XFoil with flaps deflected, output as a text file, then used within the TSFoil input file. It was found that the number of points defining each surface had to be reduced to about 70 in order for the program to run. The allowable ranges of Mach number and angle of attack are shown in Table 4.2 — numbers outside these ranges simply cause the program to close with no output generated. It was found that high subsonic freestream Mach numbers caused errors due to violation of the weak shock-wave assumption. This was as low as $M_\infty = 0.53$ for $\alpha = 9°$, and around $M_\infty = 0.75$ for $\alpha = 0°$. This violation still results in the full output but a warning is displayed at the command prompt (as seen in Fig. 4.8 for the violation occurring on all three grids), and the results are expected to be inaccurate by some degree.

Table 4.2: Allowable ranges of M and α in TSFoil

Variable	Min	Max
M	0.5	2
α	-9°	9°

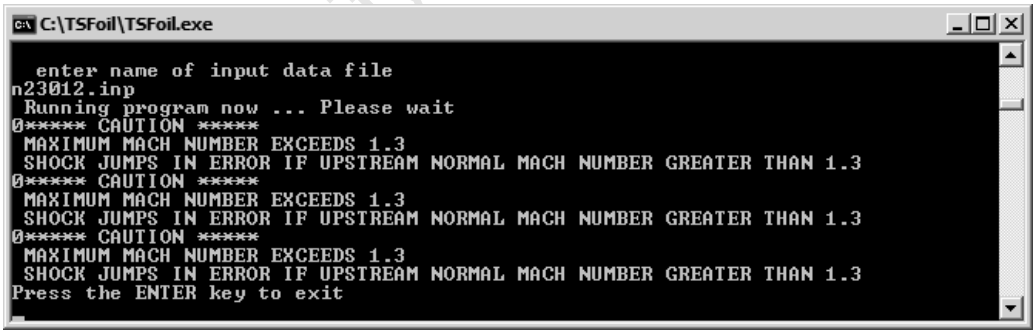


Figure 4.8: Command prompt output showing violation of the weak shock assumption

4.5.3 Preliminary Results

Some results are shown here to give a graphical example of the outputs in Appendices B.1 to B.3, which used a NACA 23012 airfoil with $\delta_a = 4°$, $\delta_t = -4°$, $\alpha = 4°$ and $M = 0.6$. The velocity field is shown in Fig. 4.9, which illustrates the grid size of the final mesh. Here the airfoil is shown in grey, with the shock wave in white. The pressure coefficient and Mach number along each surface is shown in Figs. 4.10 and 4.11 respectively, which also illustrates the airfoil discretisation.

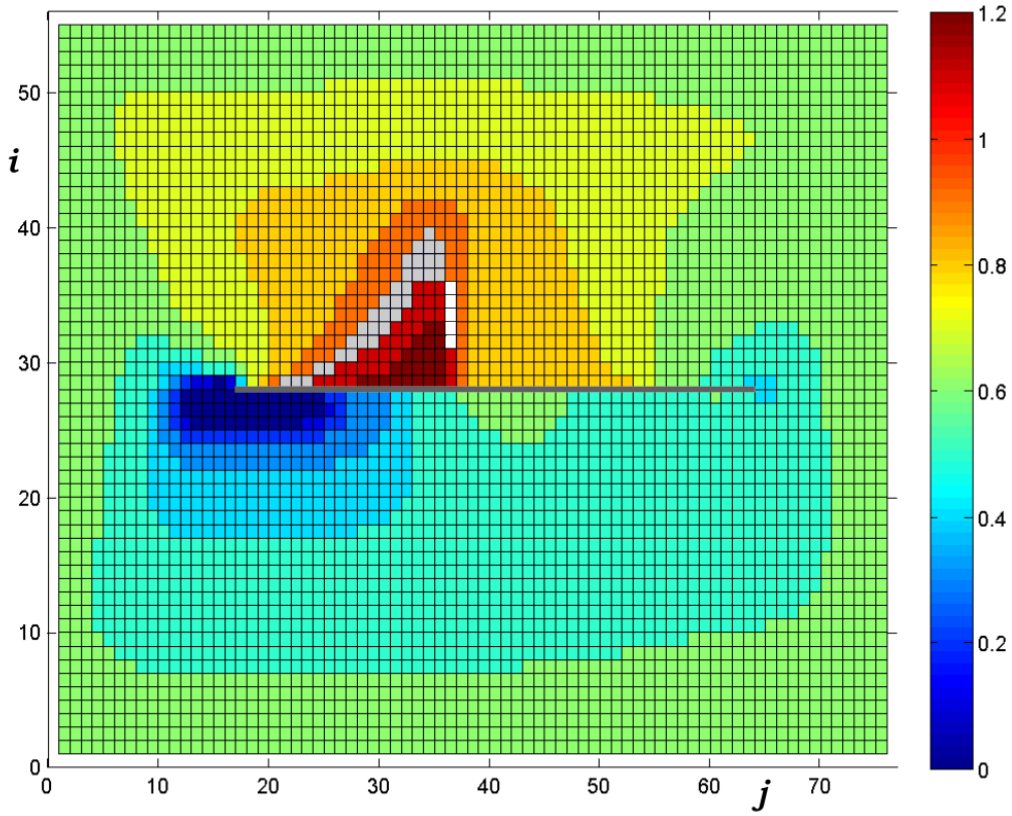


Figure 4.9: TSFoil's solution of the velocity field around a *NACA* 23012 airfoil with $\alpha = 4°$ at $M_\infty = 0.6$. Axes refer to computational cells of the final mesh, with the airfoil at $i = 28$, $27 < j < 63$. (Generated in Matlab from the text-based output.)

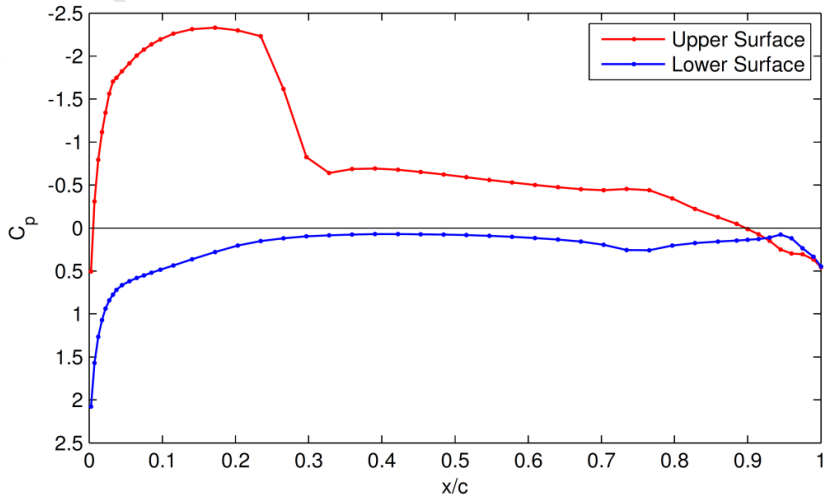


Figure 4.10: TSFoil C_p distribution over both surfaces of a *NACA* 23012 with $\alpha = 4°$ at $M_\infty = 0.6$

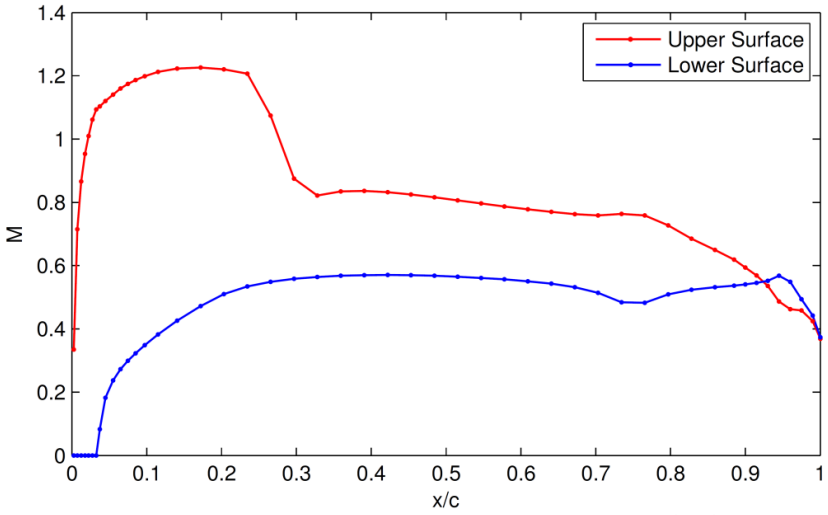


Figure 4.11: TSFoil velocity distribution over both surfaces of a *NACA* 23012 with $\alpha = 4°$ at $M_\infty = 0.6$

4.6 PABLO

PABLO was programmed by Christian Wauquiez in the late 90s at The Royal Institute of Technology in Stockholm, Sweden (known locally as the Kungliga Tekniska Högskolan, or KTH). The program was originally written to perform shape optimization of low speed airfoils, but is available for download in a form used for pedagogical reasons, where it is used as part of a course offered at the Institutionen för Flygteknik (Division of Aerodynamics) at KTH. PABLO uses a choice of elementary flows within a panel method to calculate the external flow, and an integral-method boundary-layer analysis for drag prediction. On its website[1] PABLO is sub-headed by the text

***P**otential flow around **A**irfoils with **B**oundary **L**ayer coupled **O**ne-way*

That is, PABLO uses the inviscid solution to calculate the boundary-layer parameters, but, unlike XFoil's boundary layer formulation, the inviscid solution remains unaffected by the boundary layer itself.

Although PABLO was modified for use in this project, the following sections present a brief overview of its original features. A detailed description of the modifications undertaken is left to Chapter 6.

4.6.1 Features of PABLO

Figs. 4.12 and 4.13 show some solutions for the pressure distribution over a NACA 4412 airfoil at $\alpha = 0°$. Figure 4.12 illustrates the differences between using each of the available elementary flow distributions, where C_L, C_M and x_{cp} are displayed for these purely inviscid solutions. The inaccuracy of the source distribution is clearly seen, due to its inability to model flows with circulation. Figure

[1] `http://www.nada.kth.se/~chris/pablo/pablo.html`

4.13 shows a viscous solution, which also displays C_D and points of transition and separation.

It can be seen in Figs. 4.12 and 4.13 that as well as changing the flowfield parameters, there are three ways of inputting an airfoil profile – an ellipse where the axis ratio $\frac{a}{b}$ is required, a NACA 4-digit series airfoil, and a "library file" which would contain profile data.

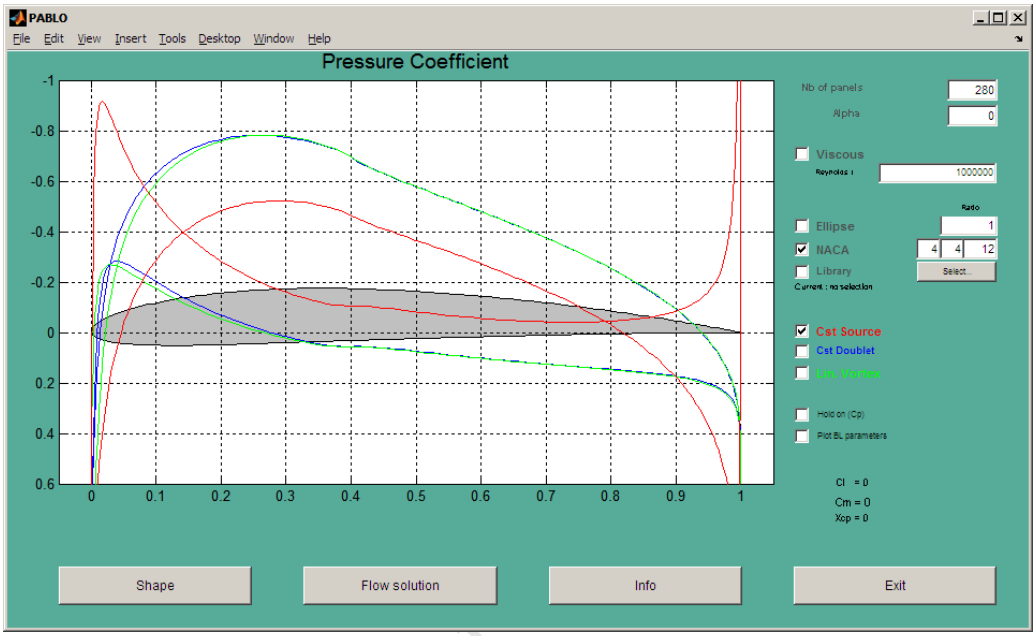


Figure 4.12: PABLO GUI showing inviscid solutions for flow over a *NACA* 4412 airfoil using a source (red), vortex (green) and doublet (blue) distribution.

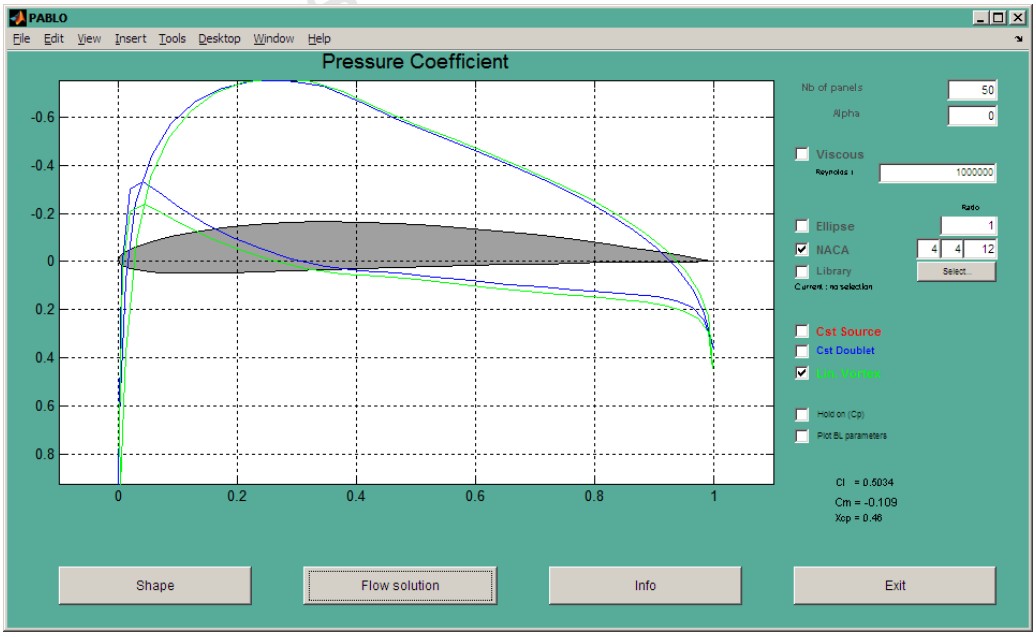


Figure 4.13: PABLO GUI showing viscous solution for flow over a *NACA* 4412 airfoil using a vortex (green) and a doublet (blue) distribution

An option is also given to plot boundary-layer parameters. The parameters displayed are δ^*, θ, H, and C_f for each surface.

4.6.2 Analysis

This section summarised from [65]

PABLO uses either a source, vortex, or doublet distribution, with the Kutta condition chosen for linearly varying singularity distributions (see §1.3 of [65] for details). The inviscid flow solver provides the tangential velocity distribution, u_e, on the airfoil's surface, and the pressure distribution is then computed from the velocity using the Bernoulli equation. The lift and moment coefficients are calculated by integrating the pressure over the body surface.

The boundary layer formulation calculates the laminar boundary layer parameters using Thwaites' method, transition is calculated using Michel's criterion, and Head's method is used for the turbulent part of the boundary layer.

4.7 CFD

4.7.1 Introduction

Along with the programs discussed in the preceding sections, a CFD analysis will be performed as a kind of benchmark. This section gives a description of the mesh, the input parameters to the solver, and includes a brief validation to ensure the results are acceptable.

4.7.2 Setting Up – I: The Mesh

The first stage of the CFD analysis involved setting up a mesh. For this ANSYS ICEM CFD v12.1 was used[2]. A C-grid was created around the airfoil with three main blocks of varying resolution, so that near the airfoil a fine grid could be used, with a coarse grid in the farfield. This can be seen in Figs. 4.14–4.16, where the mesh is shown in various levels of zoom. Figs. 4.17 and 4.18 show close-ups of the leading and trailing edges respectively, where the wall-bounded cell height was adjusted so that the $y+$ values ranges between about 35 and 580 in order to satisfy the requirements of the turbulence model used (as discussed in §3.7.1).

[2] See `http://www.ansys.com/Products/Other+Products/ANSYS+ICEM+CFD`

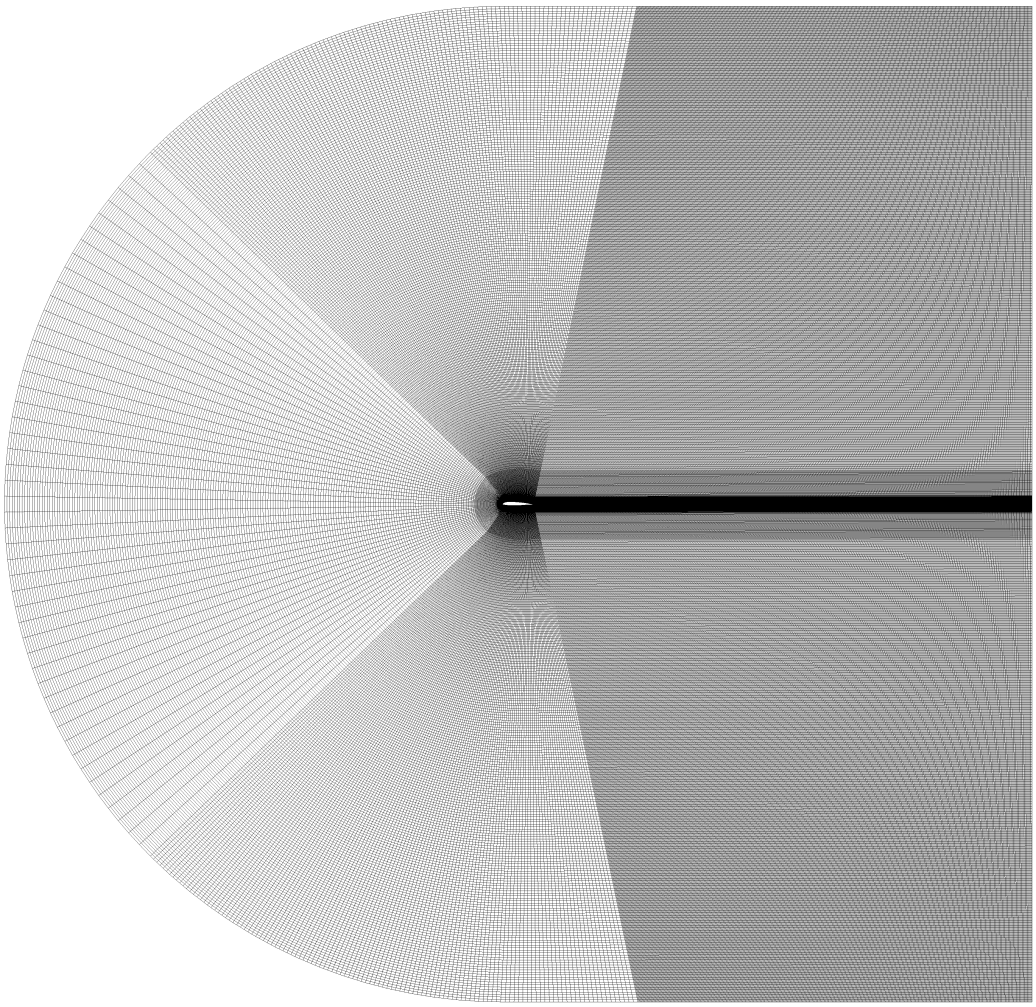


Figure 4.14: View of complete grid, where the distance from the airfoil to each farfield boundary is approximately 16c.

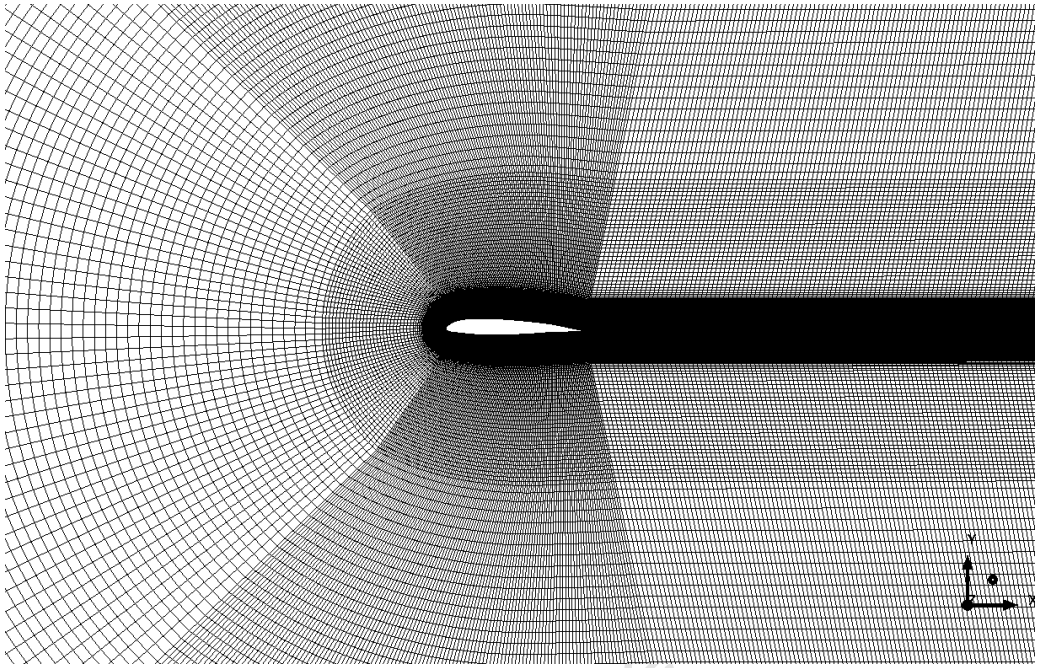


Figure 4.15: View of interior of mesh, where the three distinct blocks can clearly be seen.

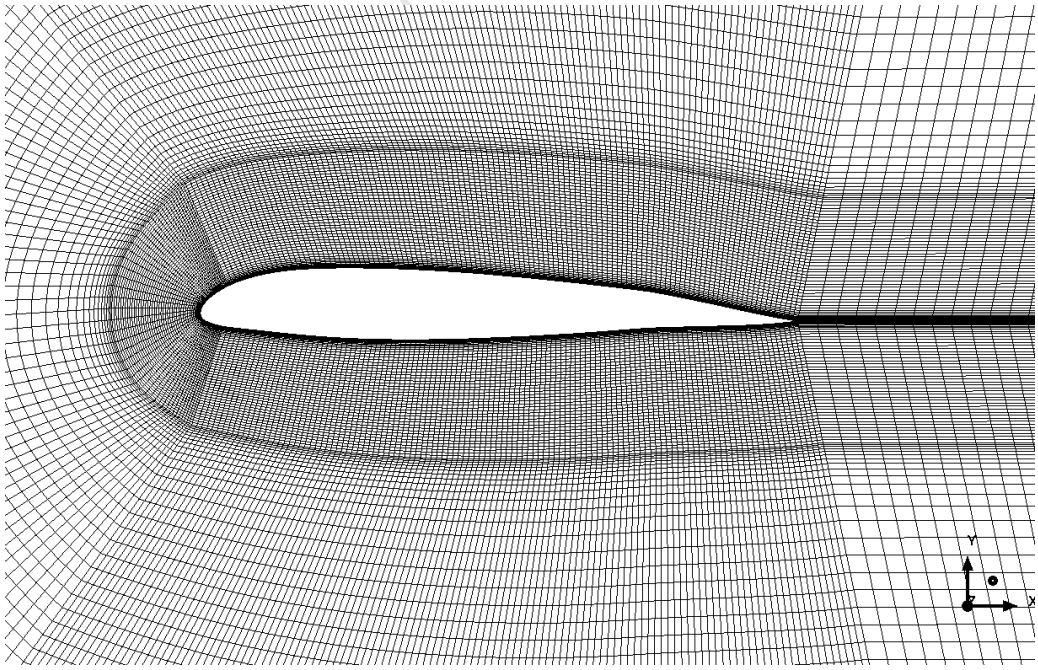


Figure 4.16: View of the innermost block and the "boundary layer" block, used mainly to tweak the $y+$ value

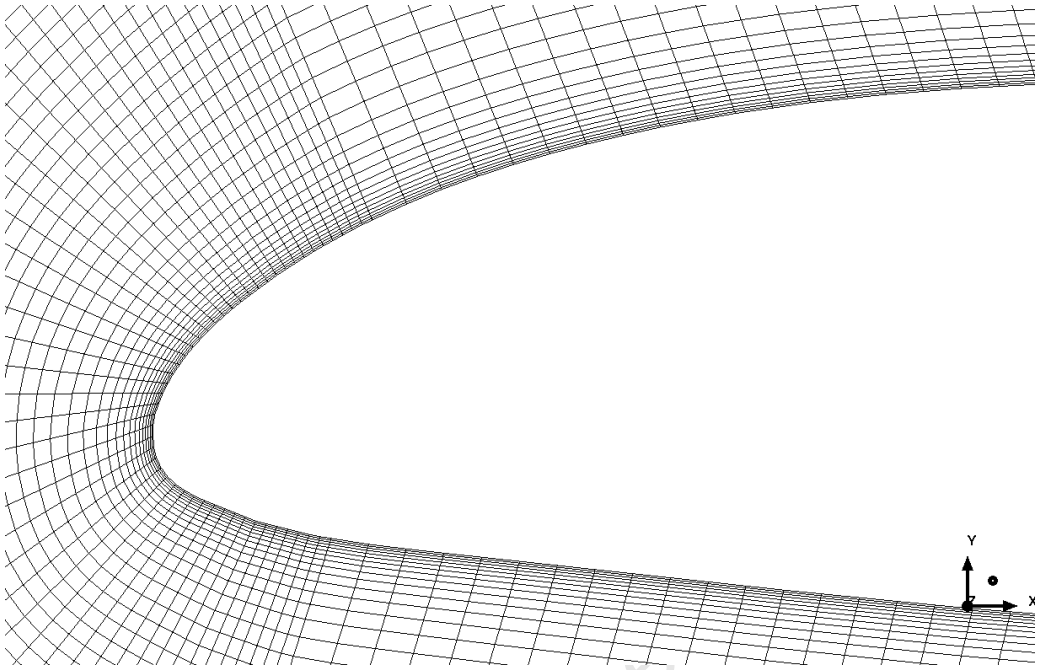


Figure 4.17: Close-up view of the leading edge, showing the wall-bounded cell sizes

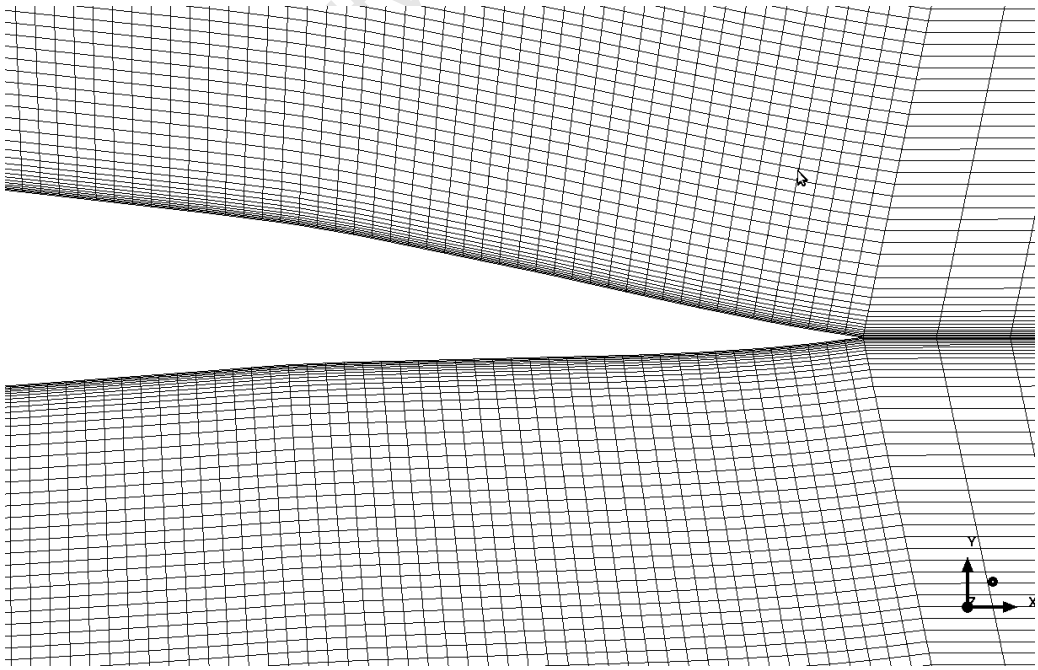


Figure 4.18: Close-up view of the trailing edge, showing the wall-bounded cell sizes

4.7.3 Setting Up – II: The Solver

The second stage involved setting up the solver to calculate the fluid flow around the airfoil. For this ANSYS FLUENT v12.1.4 was used[3]. A tutorial[4] published by Fluent Inc. was followed quite closely in terms of the parameters adjusted in the solver. Values relating to flow conditions were entered as calculated in Appendix C.

A description of these parameters is found below. The bold headings show the pane under which each setting is to be found, an example of which can be seen in Fig. 4.19. The full menu is shown in the left pane, where the "General" option has been selected, and is therefore shown in the right pane. Each of the edited options are described in the text. The order does not follow the main menu pane, but instead the order in which the parameters are entered into the solver.

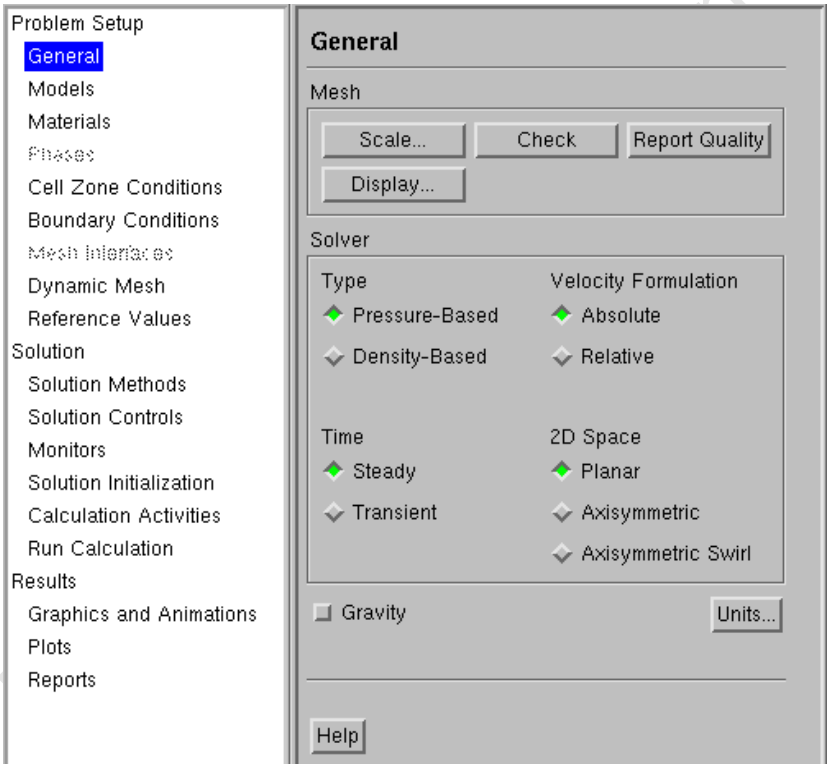


Figure 4.19: Menu pane in fluent, showing the options under Problem Setup ⇒ General

General:

Solver Type ⇒ Pressure-Based

Velocity Formulation ⇒ Absolute

Time ⇒ Steady

2D Space ⇒ Planar

[3] See `http://www.ansys.com/Products/Simulation+Technology/Fluid+Dynamics/ANSYS+FLUENT`

[4] See `http://my.fit.edu/itresources/manuals/fluent6.3/help/pdf/tg/tut03.pdf`

Models:

Energy Equation ⇒ On

Viscous ⇒ Spalart-Allmaras

Spalart-Allmaras Options ⇒ 1 Equation, Strain/Vorticity-Based, Low-Re Damping, Model Constants: $C_{b_1} = 0.1355$, $C_{b_2} = 0.622$, $C_{v_1} = 7.1$, $C_{w_2} = 0.3$, $C_{w_3} = 2$, $C_{prod} = 2$, Pr = 0.667

Material:

Type ⇒ Fluid

Material ⇒ Air

Density ⇒ Ideal Gas

C_p ⇒ 1006.43 $J/kg\,K$ (Constant)

Thermal Conductivity ⇒ 0.0242 $W/m\,K$ (Constant)

Viscosity ⇒ Sutherland Power Law ⇒ Three coefficient method, where $\mu_0 = 1.716 \times 10^{-5}\ kg/ms$, $T_0 = 273.11\,°K$, $S = 110.56\,°K$

Molecular Weight ⇒ 28.966 g/mol (Constant)

Boundary Conditions:

Gauge Pressure ⇒ $-74825\ Pa$

(see equation C.1) Mach Number ⇒ *Varied*

x-component of flow direction ⇒ *Varied*

y-component of flow direction ⇒ *Varied*

Temperature ⇒ 225

Reference Values: (these are computed from the farfield parameter settings, and are listed here where not previously stated)

Area ⇒ 1 m^2

Density ⇒ 0.4103213 kg/m^3

Depth ⇒ 1 m

Enthalpy ⇒ $-44707.59\ J/kg$

Length ⇒ 1 m

Velocity ⇒ *Varied*

Ratio of Specific Heats ⇒ 1.4

Solution Methods:

Sheme ⇒ Coupled

Gradient ⇒ Least Squares Cell Based

Pressure ⇒ Standard

Density ⇒ Second Order Upwind

Momentum ⇒ Second Order Upwind

Modified Turbulent Viscosity ⇒ Second Order Upwind

Energy ⇒ Second Order Upwind

Solution Controls:

Courant Number ⇒ 200

Explicit Relaxation Factors ⇒ Momentum ⇒ 0.5

⇒ Pressure ⇒ 0.5

Under-Relaxation Factors ⇒ Density ⇒ 0.5
⇒ Body Forces ⇒ 1
⇒ Modified Turbulent Viscosity ⇒ 0.9
⇒ Turbulent Viscosity ⇒ 1
⇒ Energy ⇒ 1

4.7.4 Validation

To ensure a degree of accuracy of the CFD results, a comparison was made with an in-house Fluent simulation which had itself been conducted for validation purposes, and had been compared with experimental results. To this end, Mach contours were compared, as well as the surface pressure distribution, C_p, and skin friction coefficient, c_f.

Table 4.3 describes the parameters used for the original validation. Here a quadrilateral mesh with a farfield boundary 10 c from the airfoil was used, a close-up of which is shown in Fig. 4.20, while the computation also utilised the Spalart-Allmaras turbulence model.

Table 4.3: Fluid properties for Fluent validation

Property	**Value**	**Units**
Mach Number, M	0.73	
Angle of Attack, α	2.80	degrees
Static Pressure, P_0	43765	Pa
Freestream Temperature, T_∞	300	K

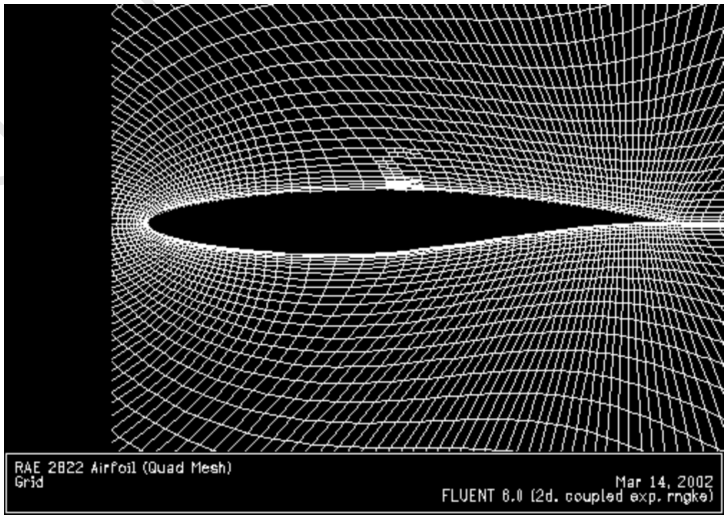


Figure 4.20: Quadrilateral mesh used by Fluent for validation purposes

Although the Mach contours given in Figs. 4.21 and 4.22 are not easily directly comparable, it can be seen that they are quite similar in nature as well as in value. The pressure coefficient seen in Fig. 4.23 is satisfactorily accurate, while

although the skin friction coefficient is not a very good match, it seems to fit the experimental data just as well as the original Fluent validation itself.

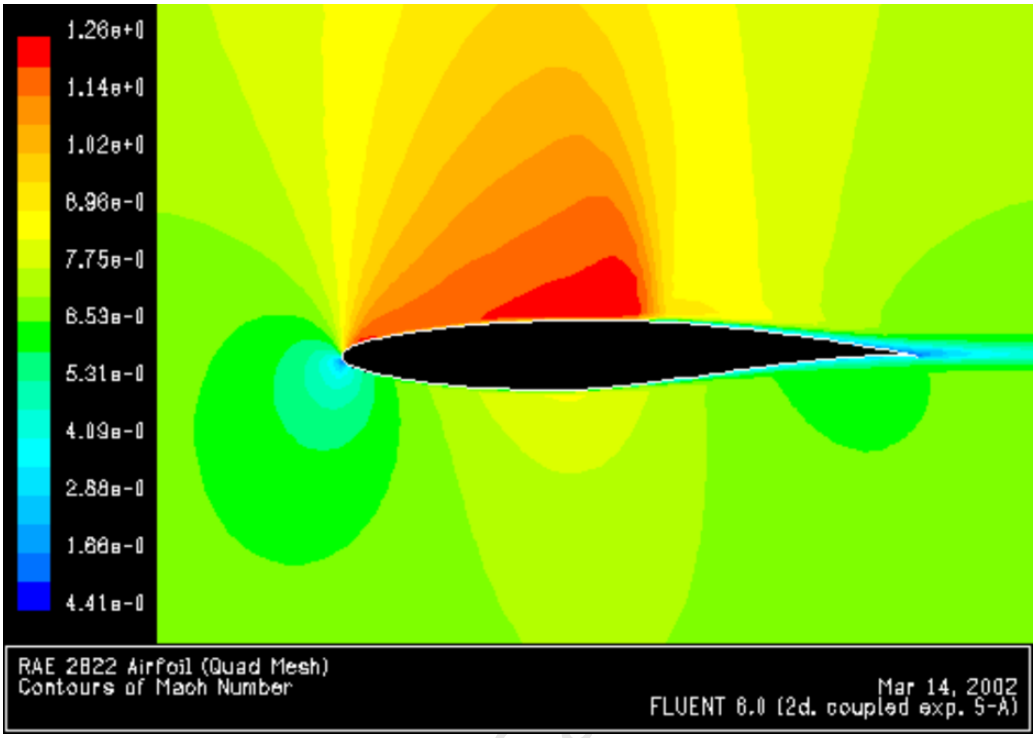


Figure 4.21: Mach distribution over RAE 2822 airfoil at $\alpha = 2.8°$, $M = 0.73$, from original Fluent validation document

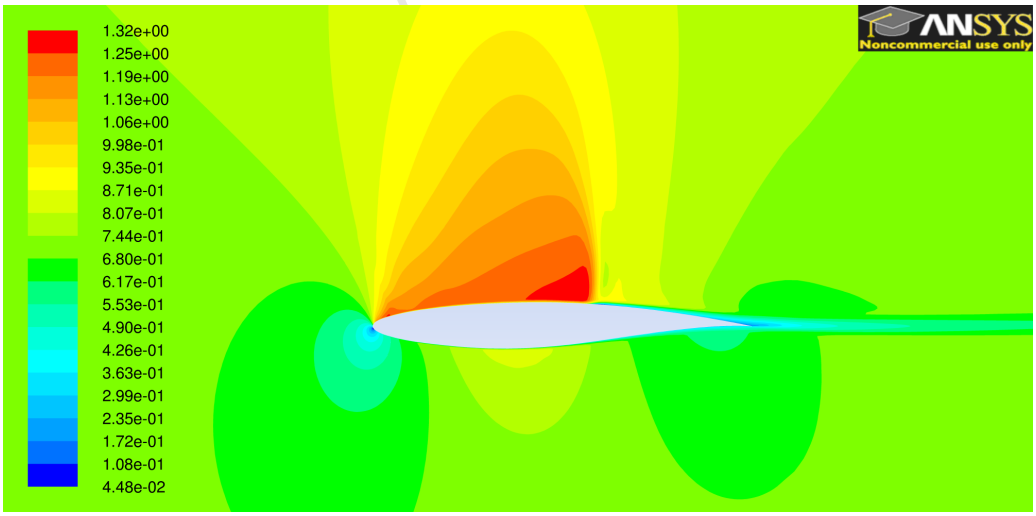


Figure 4.22: Mach distribution over RAE 2822 airfoil at $\alpha = 2.8°$, $M = 0.73$, as calculated by the method used throughout this dissertation

A closer look at Fig. 4.23[5] shows the effect of the shock wave seen in the pressure distribution over the upper surface, where an abrupt change is experienced around $\frac{x}{c} = 0.55$. Both simulations agree quite closely with the experimental data.

[5] In these figures the original Fluent data is labeled "Fluent Validation", while the data acquired by the author is labeled "Fluent Simulation".

Figure 4.24 illustrates a substantial deviation from Fluent's validation results. Here, though, the experimental data is quite sparse, and does not appear to disagree any more than the data collected by the author.

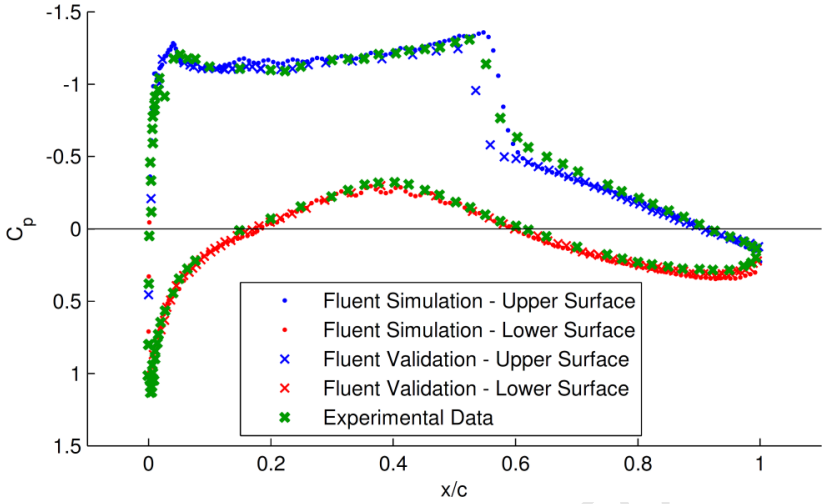


Figure 4.23: Pressure coefficient comparison with Fluent validation and experimental data

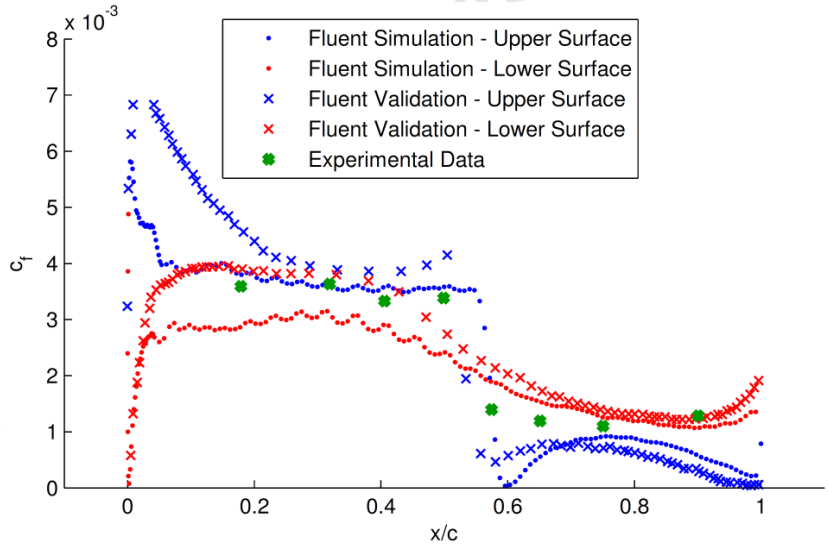


Figure 4.24: Skin friction coefficient comparison with Fluent validation and experimental data

5 Aerodynamic Analyses

5.1 Introduction

In this chapter we embark on some analyses using the software described in the previous chapter. The aim is to investigate the effect that the simplifying assumptions of each model have on how features of the flow are captured. This is done by comparing the distribution of pressure coefficient over the airfoil surface. To enable a direct comparison, and to see how the flow is modeled over the double flap, a NACA 23012 airfoil with $\delta_a = 4°$ and $\delta_t = -4°$ is used exclusively. Freestream flow conditions are as described in Appendix C.

XFoil provides the most basic purely inviscid analysis. XFoil's viscous analysis superimposes its results on a compressibility-corrected inviscid solution, as seen in Fig. 4.5, which isolates the effects of the viscous boundary layer. The effects of applying the compressibility correction are therefore investigated separately. Thereafter the application of the boundary-layer equations is investigated.

to examine the effects of shock waves. The analysis is inviscid and includes small disturbance compressibility effects, so is compared with XFoil's inviscid compressibility-corrected results.

The CFD analyses are then directly compared with the potential analyses. Assuming the CFD analysis to be valid, this will give an indication of how closely the potential theories match reality.

5.2 Simulations Performed

A full range of conditions could not always be tested, as each method had a slightly different range of applicability. Table 5.1 shows the list of simulation runs that were completed. The specific reasons for the limits on these are varied, but all are due to the inability of each solver to model the flow for parameters just outside these ranges. XFoil's viscous analysis fails to converge at high subsonic freestream Mach numbers, and either no output is given or the output is disregarded. TSFoil has a limit on the freestream Mach number of $0.5 < M_\infty < 2$, and a limit on angle of attack of $-9° < \alpha < 9°$. A full range of α was tested and a ceiling of $M_\infty = 0.9$ was imposed. An attempt was made to match the CFD analyses to the α range of TSFoil, but convergence did not occur at $M_\infty = 0.5$ for $\alpha = -9°$ and $\alpha = 9°$. The CFD solver also failed to converge for $M_\infty = 0.3$ when α was fixed at 3°.

Table 5.1: Simulations performed for comparative purposes

Platform	Fixed	Varied	Minimum	Increment	Maximum
XFoil	$M_\infty = 0.3$	α	-10°	+5°	20°
XFoil	$\alpha = 3°$	M_∞	0.1	+0.1	0.6
TSFoil	$M_\infty = 0.5$	α	-9°	+3°	9°
TSFoil	$\alpha = 3°$	M_∞	0.5	+0.1	0.9
CFD	$M_\infty = 0.5$	α	-6°	+3°	6°
CFD	$M_\infty = 0.8$	α	-9°	+3°	9°
CFD	$\alpha = 3°$	M_∞	0.4	+0.1	0.9

A discussion of the results will appear at the end of the chapter, in §5.8 on page 71.

5.3 Application of the Kármán-Tsien Compressibility Correction

The results of an XFoil inviscid analysis at $\alpha = 3°$ is shown (black line) in Fig. 5.1 below, with the Kármán-Tsien compressibility correction applied for three different freestream Mach numbers.

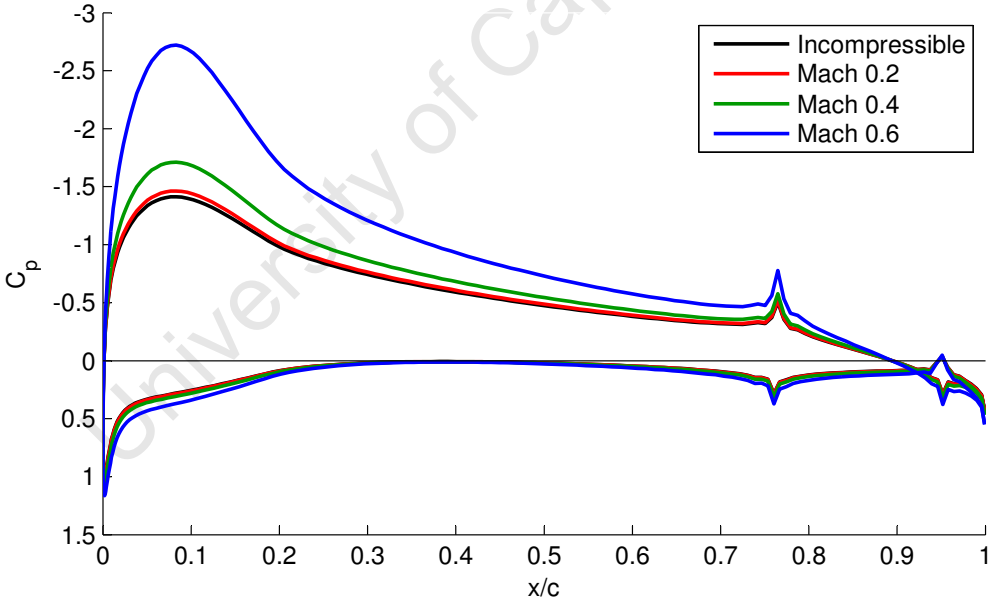


Figure 5.1: Kármán-Tsien compressibility correction at different freestream Mach numbers for a NACA 23012 airfoil with $\delta_a = 4°$, $\delta_t = -4°$ and $\alpha = 3°$

The Kármán-Tsien compressibility correction equation from §3.3.3 is repeated below for convenience. It will be seen later, that even though the equation appears quite straightforward, it is quite accurate until supersonic regions appear in the flow.

$$C_p = \frac{C_{p_0}}{\sqrt{1 - M_\infty^2} + \left[\frac{M_\infty^2}{1+\sqrt{1-M_\infty^2}}\right]\frac{C_{p_0}}{2}} \tag{3.4}$$

5.4 Comparison: Linear Potential Flow and with Viscous Boundary-Layer

The results of adding the effects of viscosity in the boundary layer were investigated by comparing XFoil's viscous and compressible inviscid results. First a Mach-sweep was computed to investigate the effects of an increase in velocity. The C_p distributions for each simulation appear in Figs. 5.2 to 5.7. Here the blue and red lines are the viscous solution over the upper and lower surfaces respectively, while the dotted line represents the inviscid solution.

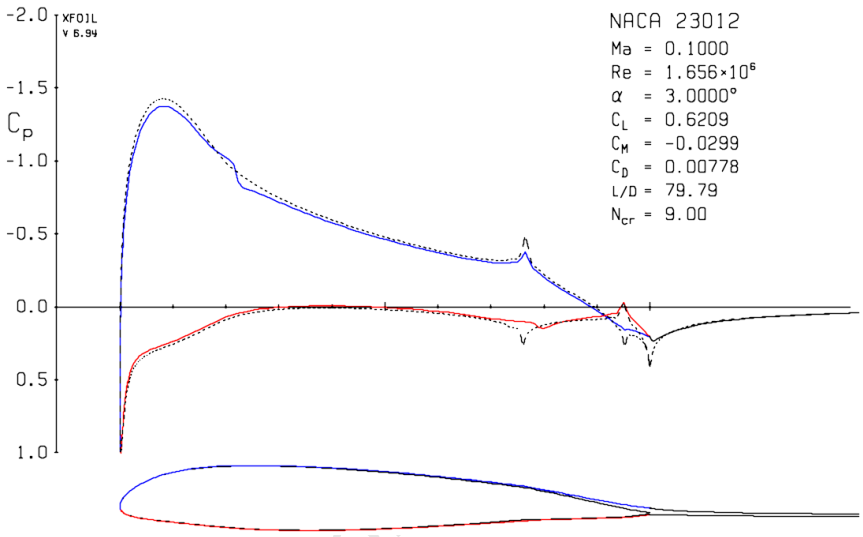


Figure 5.2: XFoil viscous vs inviscid comparison, $\alpha = 3°$, $M = 0.1$

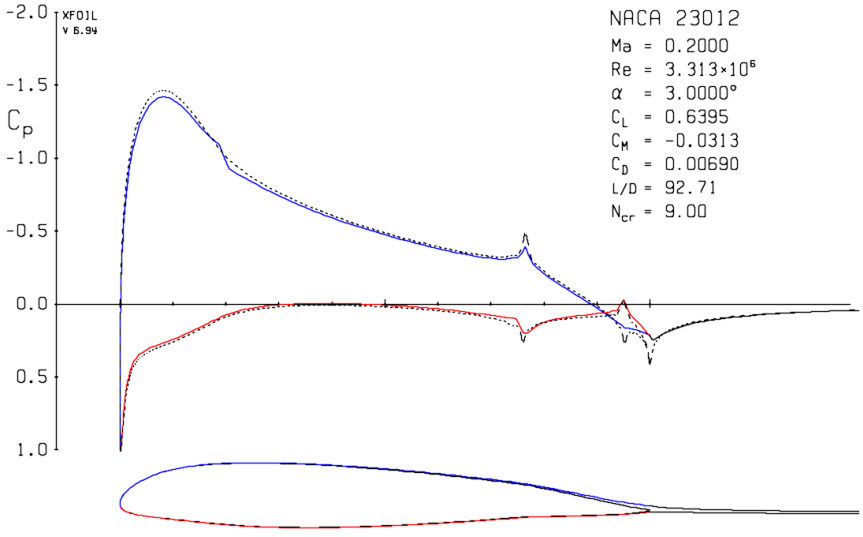


Figure 5.3: XFoil viscous vs inviscid comparison, $\alpha = 3°$, $M = 0.2$

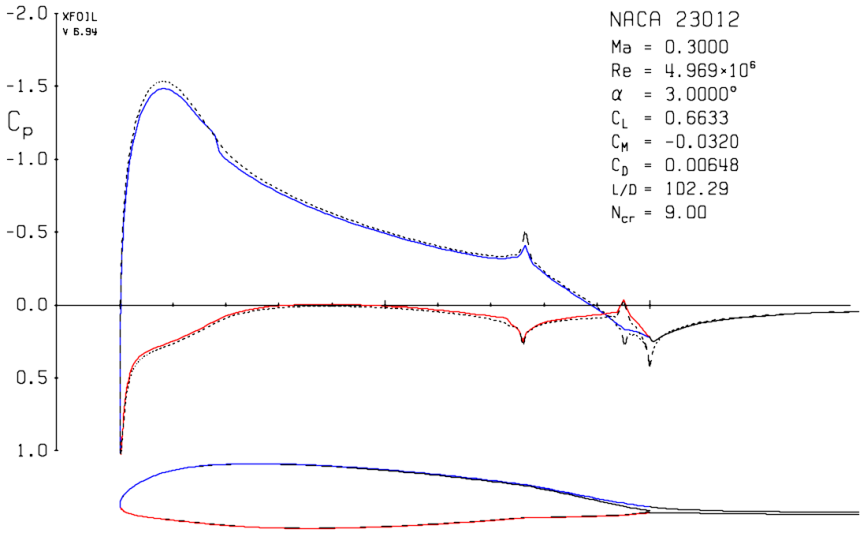


Figure 5.4: XFoil viscous vs inviscid comparison, $\alpha = 3°$, $M = 0.3$

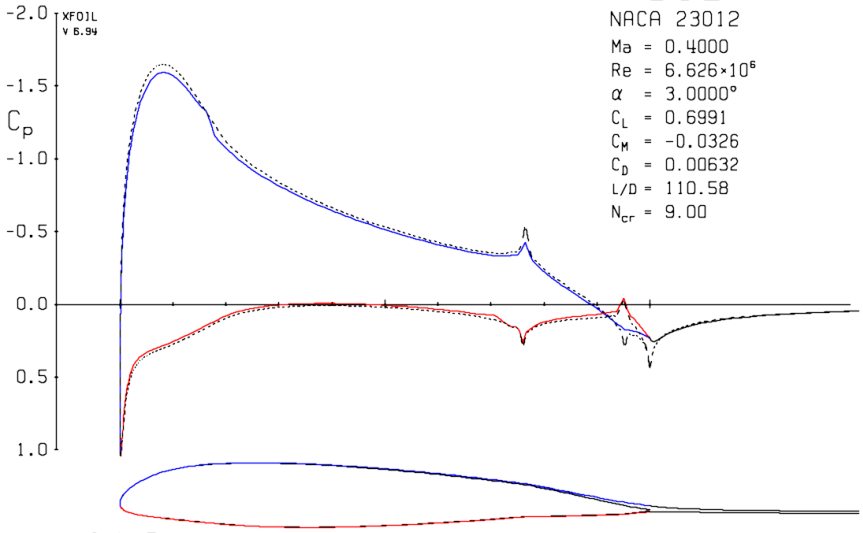


Figure 5.5: XFoil viscous vs inviscid comparison, $\alpha = 3°$, $M = 0.4$

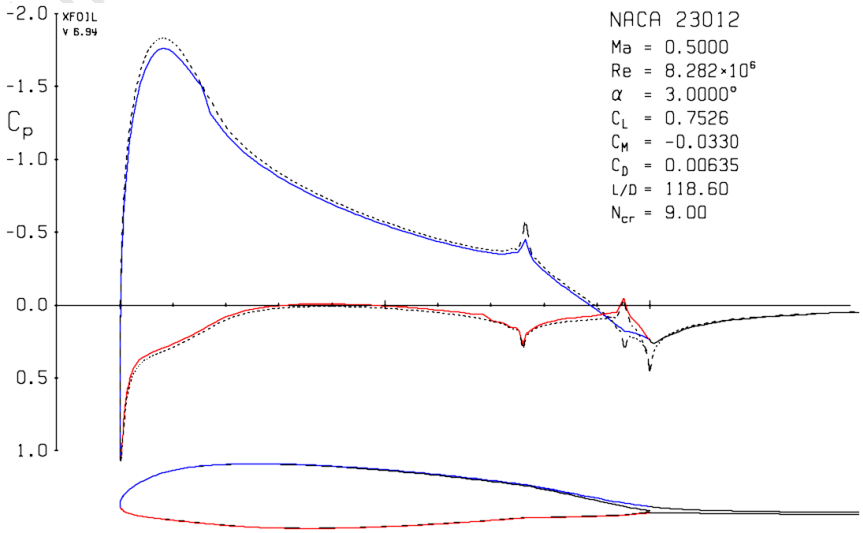


Figure 5.6: XFoil viscous vs inviscid comparison, $\alpha = 3°$, $M = 0.5$

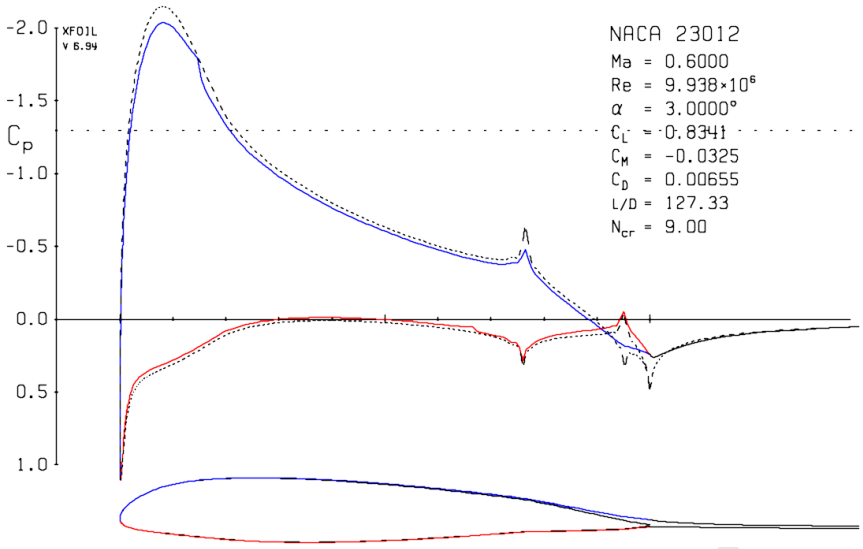


Figure 5.7: XFoil viscous vs inviscid comparison, $\alpha = 3°$, $M = 0.6$

The effects of viscosity were also investigated by performing an α-sweep for the range $-10° < \alpha < 15°$, with M_∞ fixed at 0.3. These can be seen in Figs. 5.8 to 5.13. Particularly at high angles of attack (eg Fig. 5.13) separation occurs which affects the pressure distribution to a far greater extent than seen in the previous results.

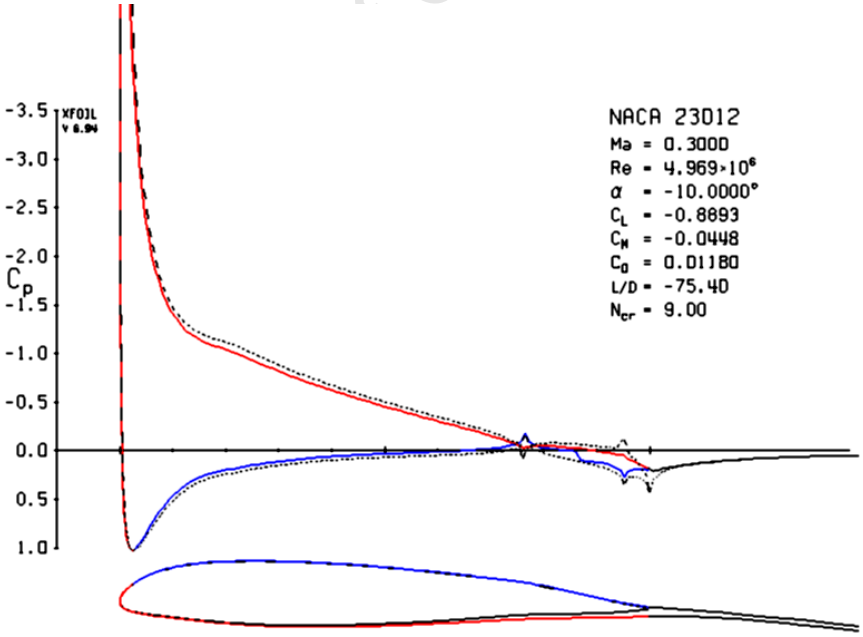


Figure 5.8: XFoil viscous vs inviscid comparison, $M = 0.3$, $\alpha = -10°$

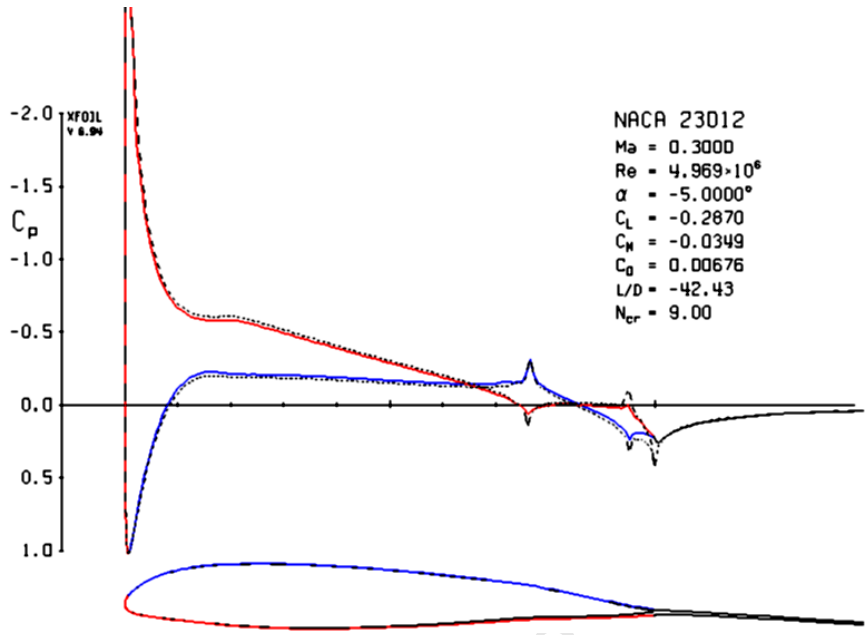


Figure 5.9: XFoil viscous vs inviscid comparison, $M = 0.3$, $\alpha = -5°$

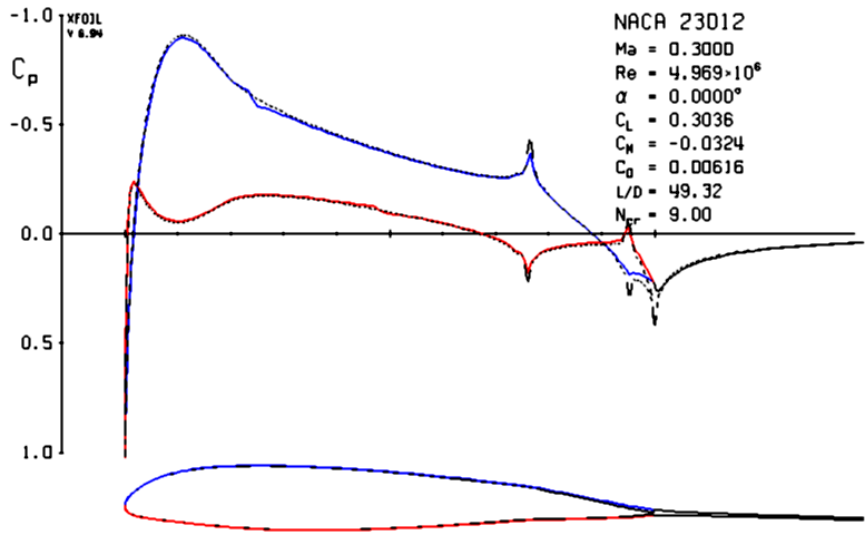


Figure 5.10: XFoil viscous vs inviscid comparison, $M = 0.3$, $\alpha = 0°$

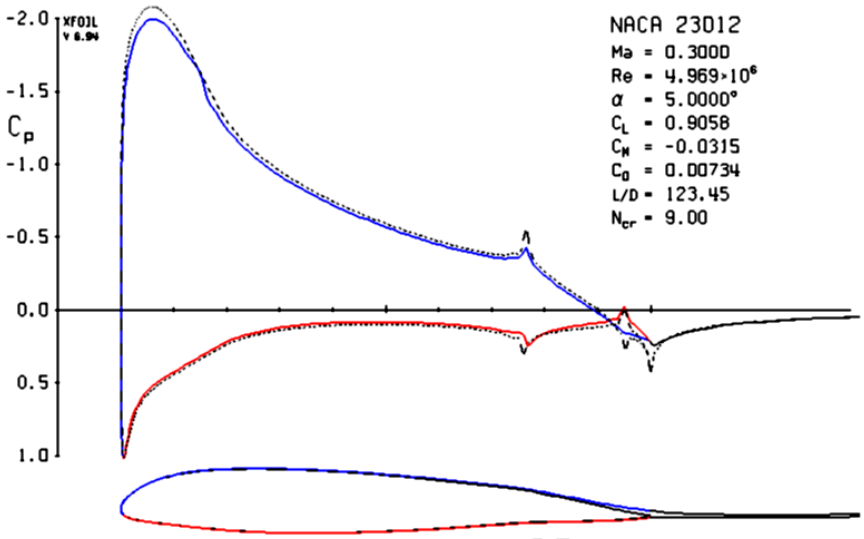


Figure 5.11: XFoil viscous vs inviscid comparison, $M = 0.3$, $\alpha = 5°$

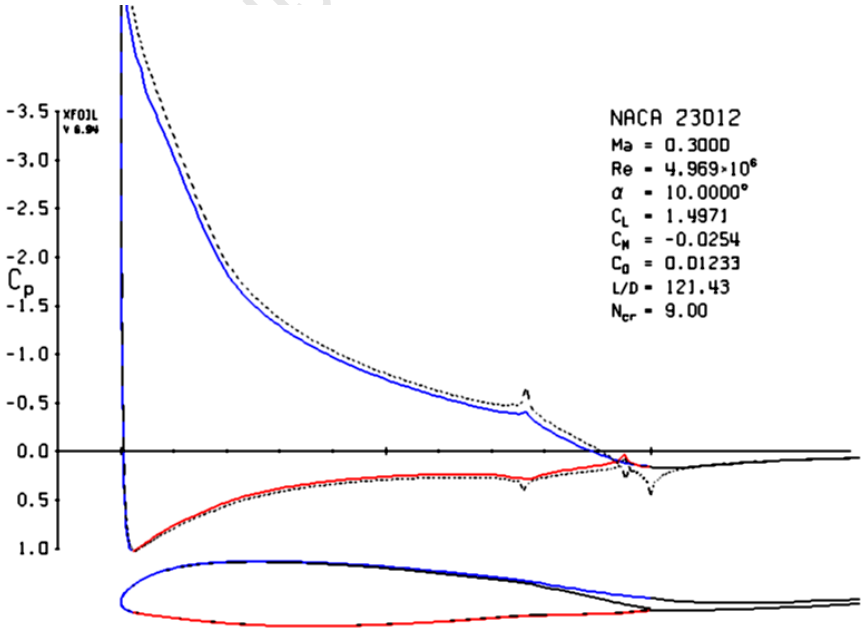


Figure 5.12: XFoil viscous vs inviscid comparison, $M = 0.3$, $\alpha = 10°$

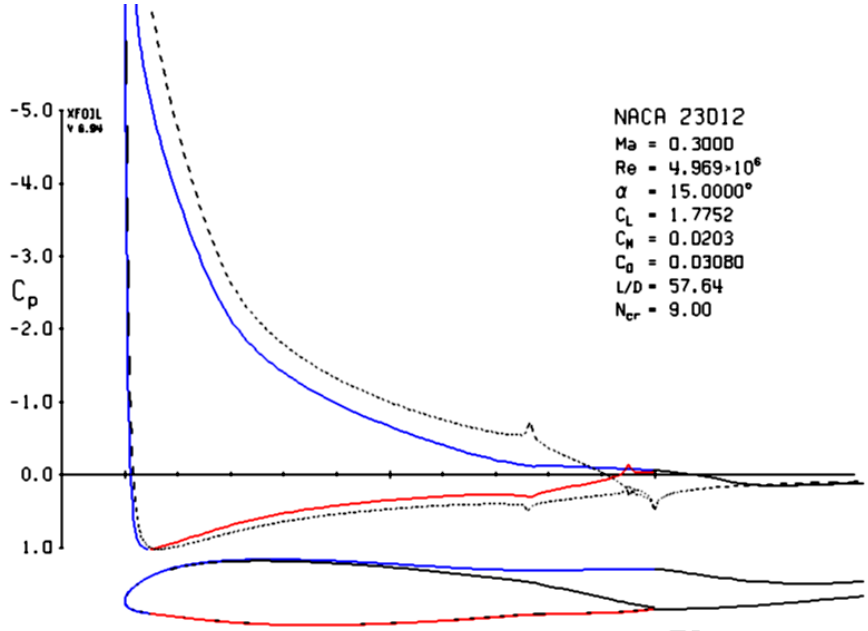


Figure 5.13: XFoil viscous vs inviscid comparison, $M = 0.3$, $\alpha = 15°$

5.5 Comparison: Non-linear and Linear Potential Flow

The inviscid transonic formulation of TSFoil is now compared with XFoil's inviscid results to investigate the effects captured by the TSD equations. Presented first are the results of a variation in Mach number. Here the Mach number was increased from 0.5 to 0.8, in increments of 0.1, with α held constant at 3°. TSFoil only allows a minimum Mach number of 0.5, hence the high starting value. The evidence of shock starts to appear at $M_\infty = 0.6$, so the XFoil components of the plot are only shown for comparative purposes. For $M_\infty = 0.7$ TSFoil gives warnings on the fine grid only that the normal Mach number exceeds 1.3, violating the weak shock assumption, and weakening the confidence in the accuracy of the result. For $M_\infty = 0.8$ these warnings are given for all three grids. These results are shown in any case for completeness. The full set of results appear in Figs. 5.14 to 5.17.

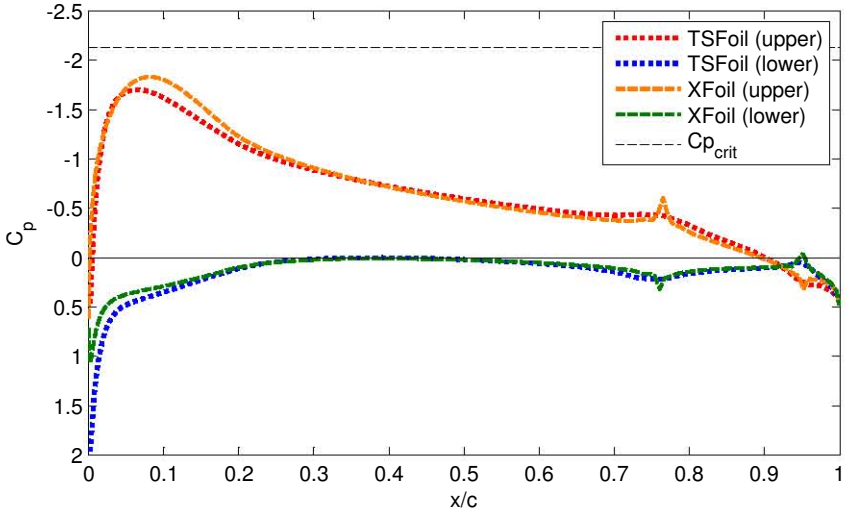


Figure 5.14: TSFoil vs XFoil inviscid formulation for $M = 0.5$, $\alpha = 3°$

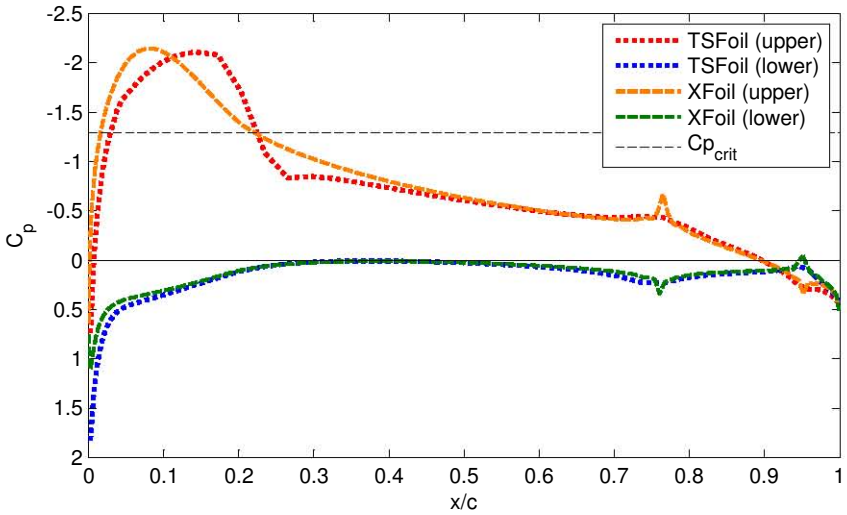


Figure 5.15: TSFoil vs XFoil inviscid formulation for $M = 0.6$, $\alpha = 3°$

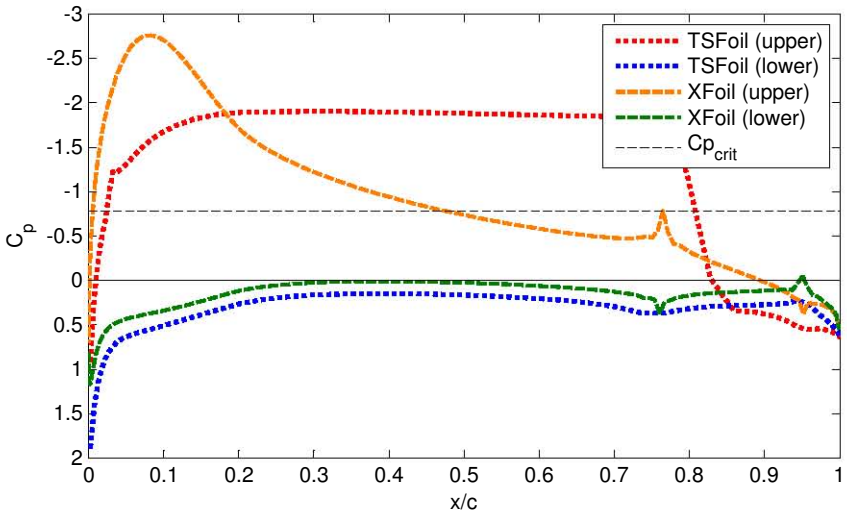


Figure 5.16: TSFoil (not converged on fine grid) vs XFoil inviscid formulation for $M = 0.7$, $\alpha = 3°$

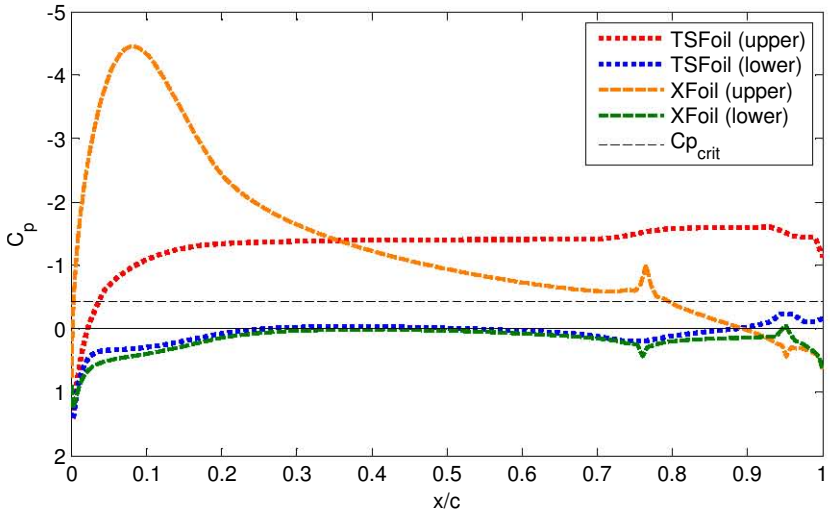


Figure 5.17: TSFoil (not converged on all three grids) vs XFoil inviscid formulation for $M = 0.8$, $\alpha = 3°$

A data set was also collected for the full range of angle of attack, where $-9° < \alpha < 9°$ incremented by 3° at a time. Here it was found that only a low Mach number of 0.5 could be used to ensure $M_n < 1.3$ throughout the flow. Figs. 5.18 to 5.24 show the results.

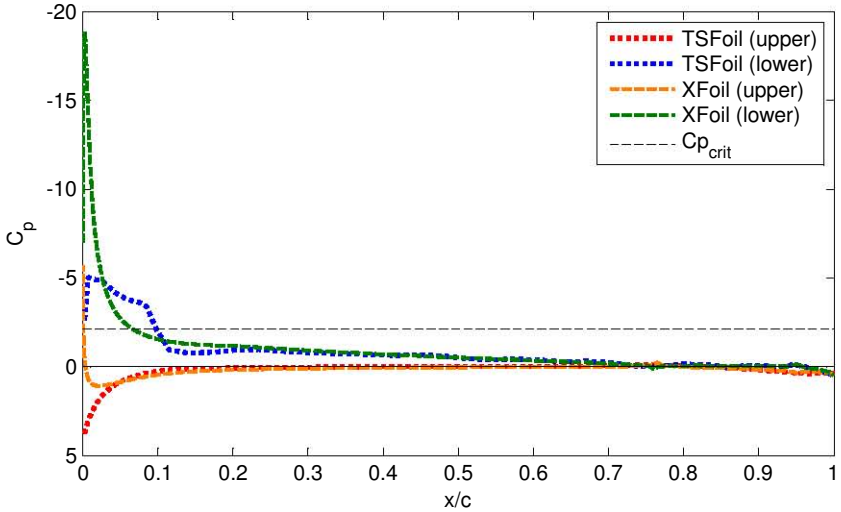


Figure 5.18: TSFoil vs XFoil inviscid formulation, $M = 0.5$, $\alpha = -9°$.

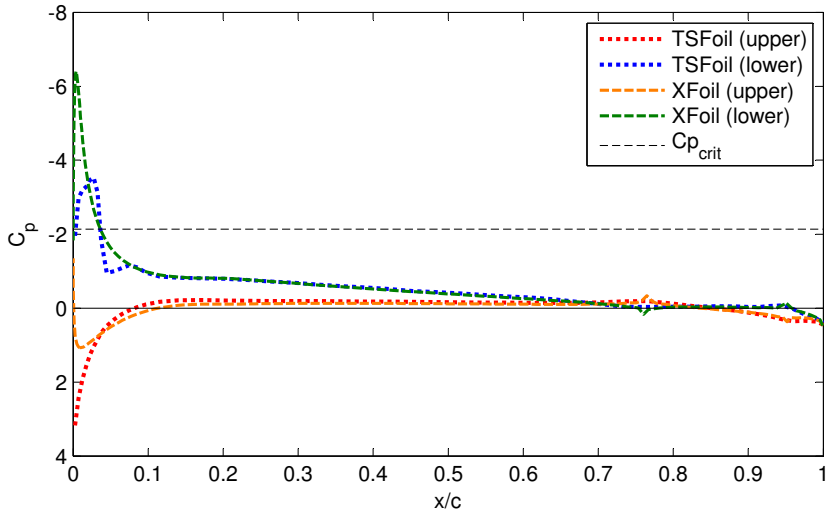


Figure 5.19: TSFoil vs XFoil inviscid formulation, $M = 0.5$, $\alpha = -6°$.

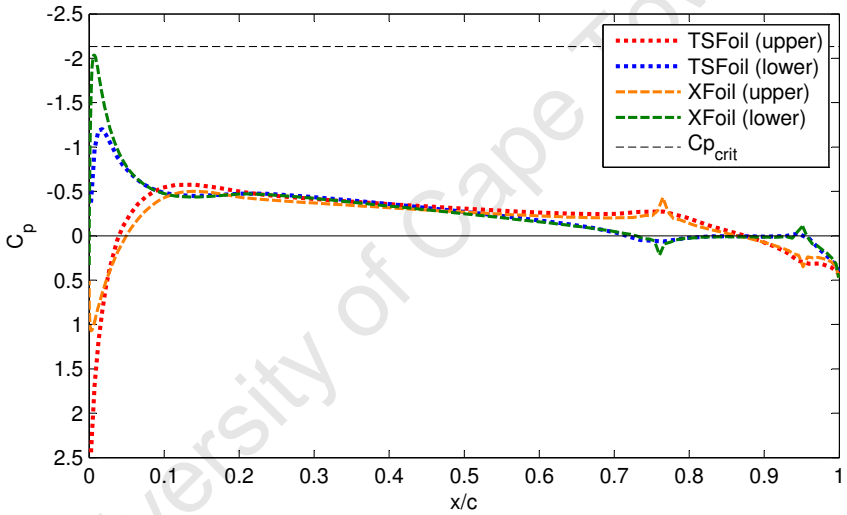


Figure 5.20: TSFoil vs XFoil inviscid formulation, $M = 0.5$, $\alpha = -3°$.

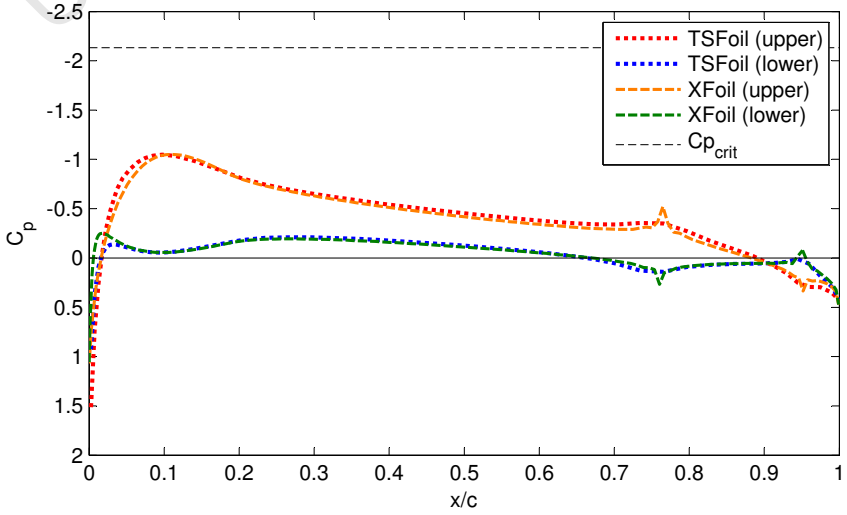


Figure 5.21: TSFoil vs XFoil inviscid formulation, $M = 0.5$, $\alpha = 0°$.

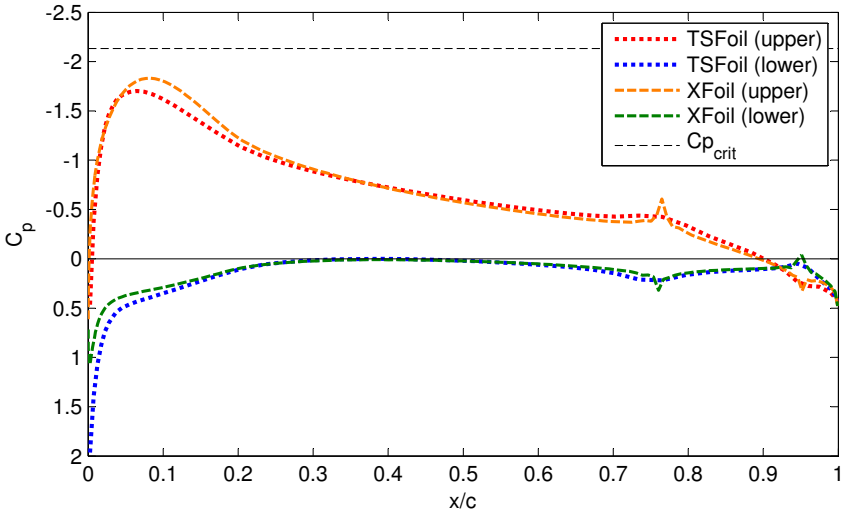


Figure 5.22: TSFoil vs XFoil inviscid formulation, $M = 0.5$, $\alpha = 3°$.

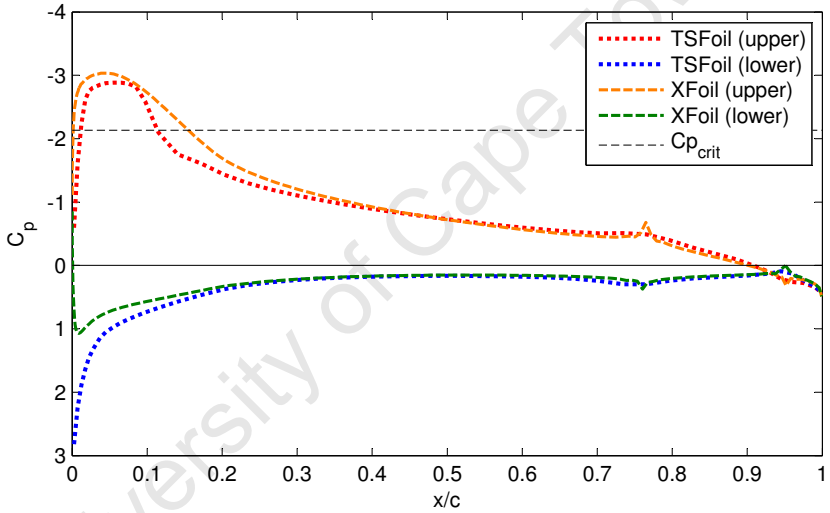


Figure 5.23: TSFoil vs XFoil inviscid formulation, $M = 0.5$, $\alpha = 6°$.

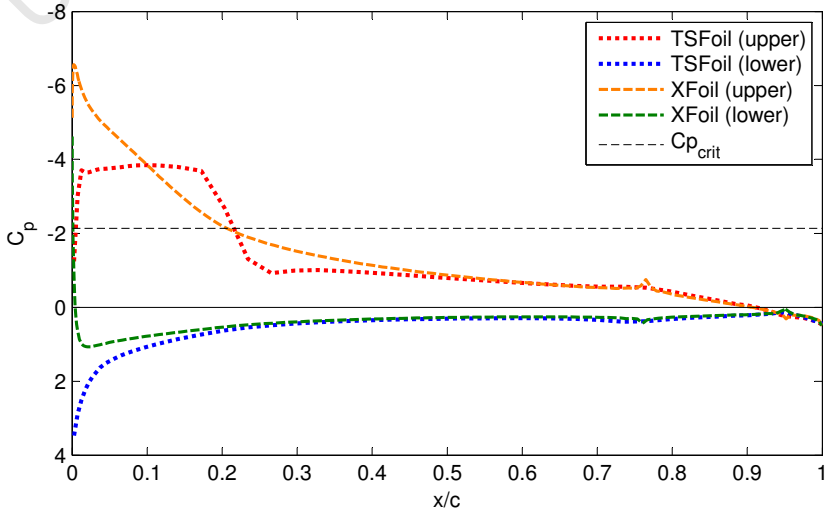


Figure 5.24: TSFoil vs XFoil inviscid formulation, $M = 0.5$, $\alpha = 9°$.

5.6 Comparison: Navier-Stokes and Linear Potential Flow with Viscous Boundary Layer

A comparison between XFoil and CFD is now presented. Here the results of the variation in velocity are presented, to illustrate XFoil's accuracy at low Mach numbers, and it's unsurprising inability to capture the behaviour of shocked flows.

Figures 5.25 and 5.26 show the C_p distribution for a NACA 23012 airfoil with $\delta_a = 4°$, $\delta_t = -4°$, $\alpha = 3°$ and at $M_\infty = 0.4$ and $M_\infty = 0.5$ respectively. Seen in Fig. 5.27 is the freestream flow at $M_\infty = 0.6$, where a portion of the flow goes supersonic and a noticeable deviation in the results starts to occur. This deviation increases when $M_\infty = 0.7$ in Fig. 5.28.

It can be seen that for the cases where the flow is everywhere subsonic XFoil's results are very similar to the CFD results, although there is slight deviation at the profile's sharp edges near the control surface hinges. The fact that the results match closely even when $M_\infty = 0.5$ in Fig 5.26 illustrates the accuracy of the Kármán-Tsien compressibility correction all the way to sonic conditions.

Figures 5.15, 5.24 and 5.27 show $C_{p_{\text{crit}}}$ intersecting the XFoil C_p distribution at the shock. However, this would seem fortuitous as Fig. 5.28 shows the shock displaced from the intersection point, indicating that the trend is not necessarily always true.

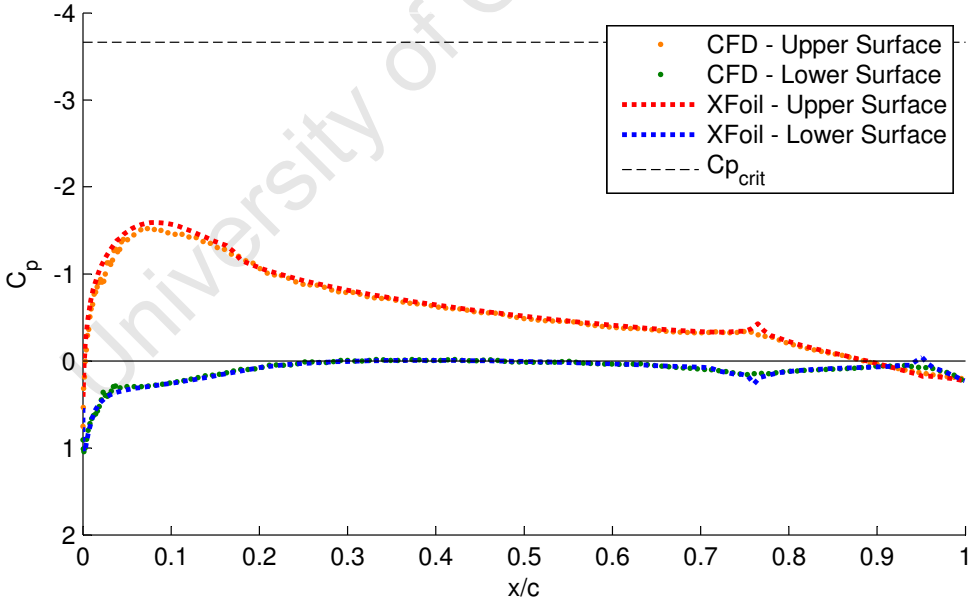


Figure 5.25: XFoil vs CFD C_p distribution for $M_\infty = 0.4$ and $\alpha = 3°$

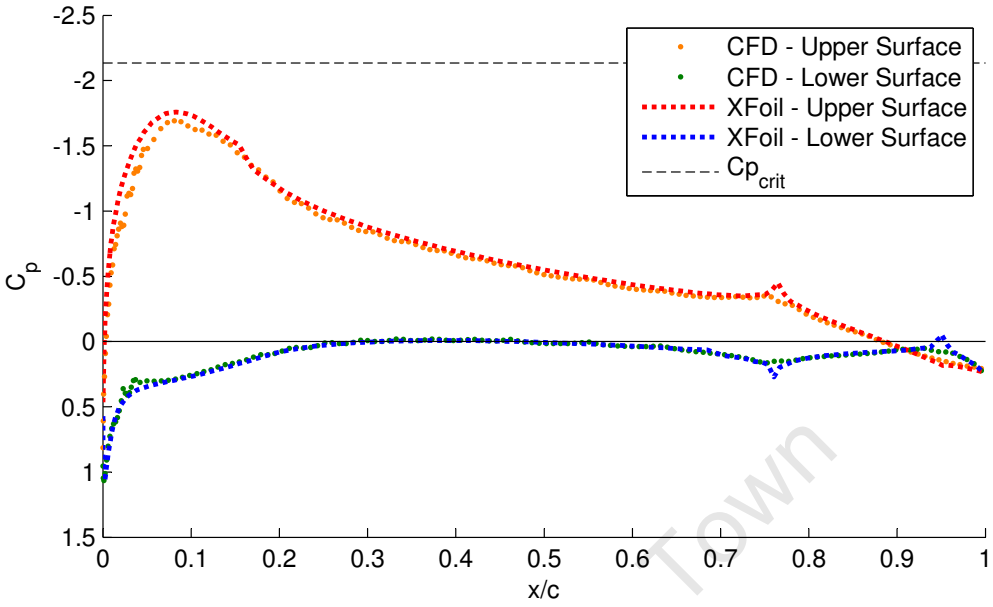


Figure 5.26: XFoil vs CFD C_p distribution for $M_\infty = 0.5$ and $\alpha = 3°$

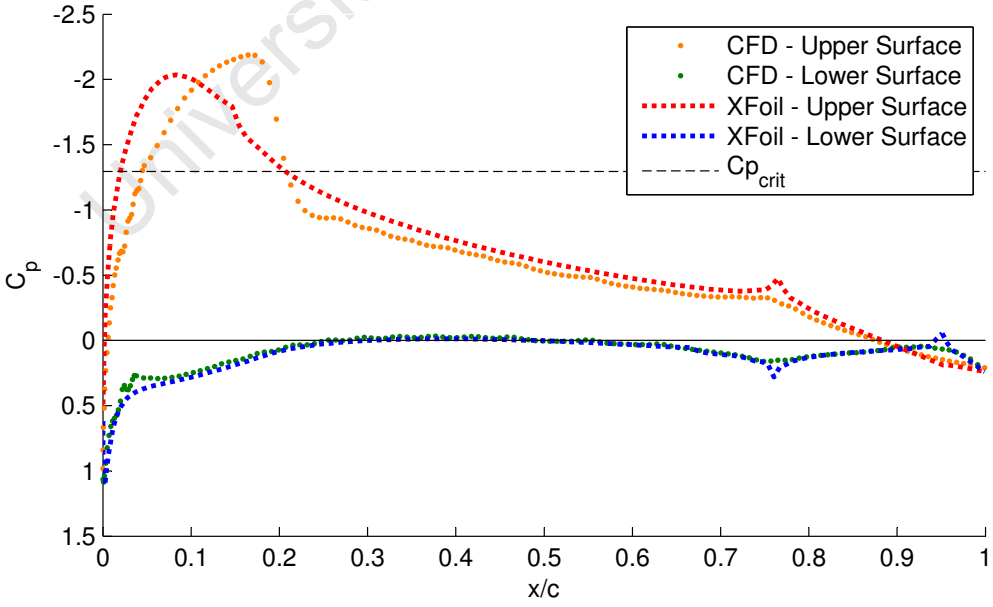


Figure 5.27: XFoil vs CFD C_p distribution for $M_\infty = 0.6$ and $\alpha = 3°$

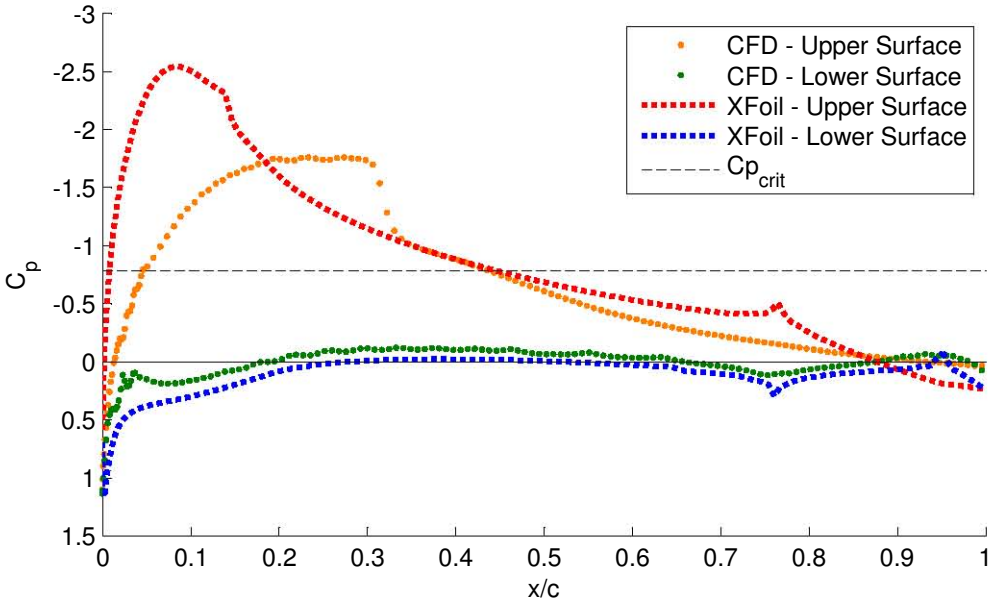


Figure 5.28: XFoil vs CFD C_p distribution for $M_\infty = 0.7$ and $\alpha = 3°$

5.7 Comparison: Navier-Stokes and Non-linear Potential Flow

The same comparison as before was made with CFD and TSFoil. Here again the NACA 23012 airfoil with $\delta_a = 4°$, $\delta_t = -4°$ was used, with an α of 3° and a variation in Mach from 0.5 to 0.9. For some of the simulations at high Mach numbers, warnings were given that the weak-shock assumption was violated – that is, for all occasions where $M_n > 1.3$. These are summarised in Table 5.2 where the affected pressure coefficient distributions are shown in Figs. 5.31 to 5.33.

Table 5.2: Summary of the TSFoil Mach-sweep simulations, where violations of weak-shock assumption were given.

Mach	**Mesh**			**Figure**
	Coarse	**Medium**	**Fine**	
0.5	valid	valid	valid	5.29
0.6	valid	valid	valid	5.30
0.7	valid	valid	violated	5.31
0.8	violated	violated	violated	5.32
0.9	violated	violated	violated	5.33

It can be seen that the results where the weak-shock assumption is valid correspond fairly well with the CFD results. However, even for the run at Mach 0.7 – where only the fine mesh gave $M_n > 1.3$, suggesting that M_n exceeded this amount only slightly – the results are nevertheless wildly different to the CFD analysis.

The results of a variation in angle of attack (Figs. 5.34–5.38 on pages 69–70) were performed at a Mach number so that M_n was always below the allowable range. Here a good correlation between the two methods can be seen throughout the whole range.

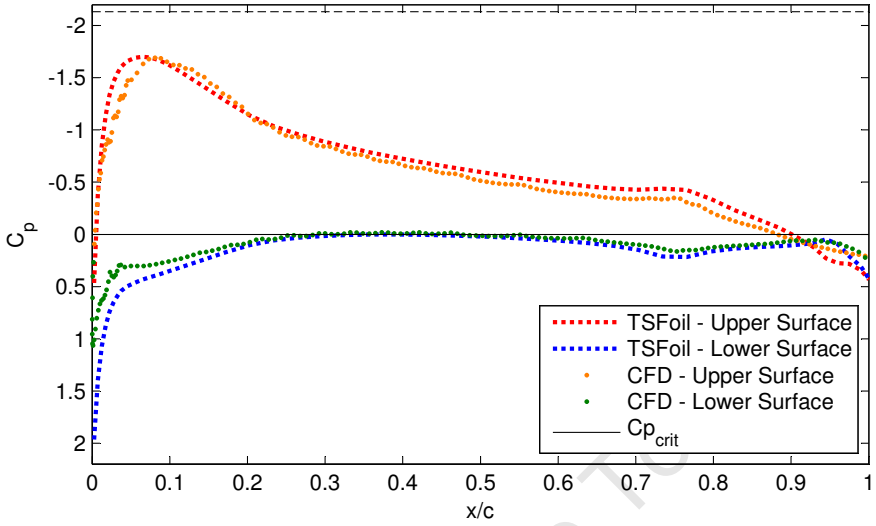


Figure 5.29: TSFoil vs CFD C_p distribution for $\alpha = 3°$, $M_\infty = 0.5$

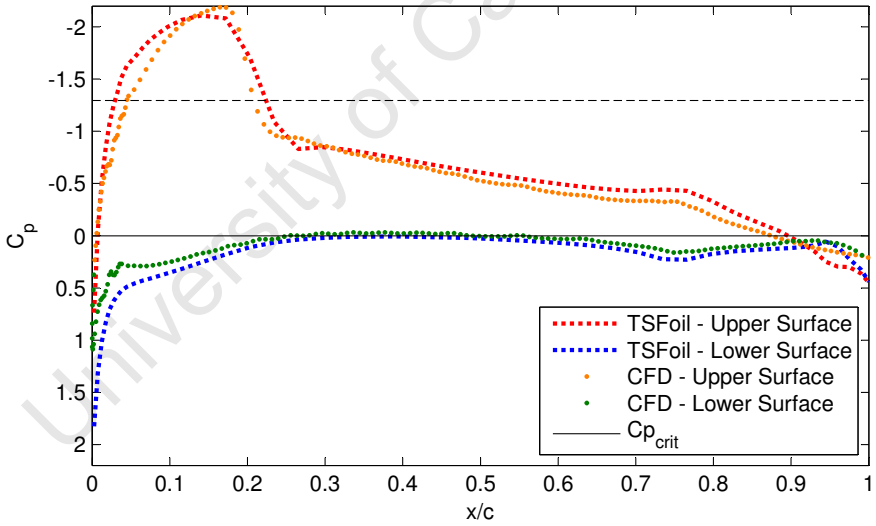


Figure 5.30: TSFoil vs CFD C_p distribution for $\alpha = 3°$, $M_\infty = 0.6$

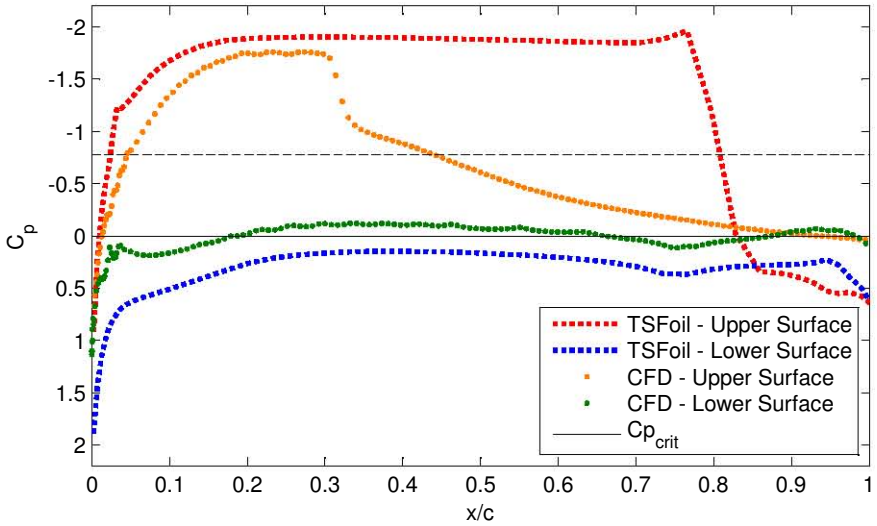


Figure 5.31: TSFoil vs CFD C_p distribution for $\alpha = 3°$, $M_\infty = 0.7$

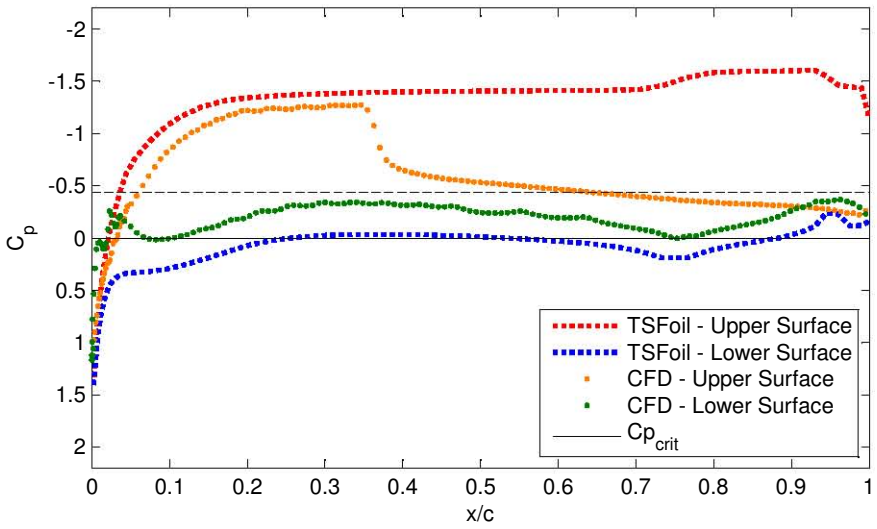


Figure 5.32: TSFoil vs CFD C_p distribution for $\alpha = 3°$, $M_\infty = 0.8$

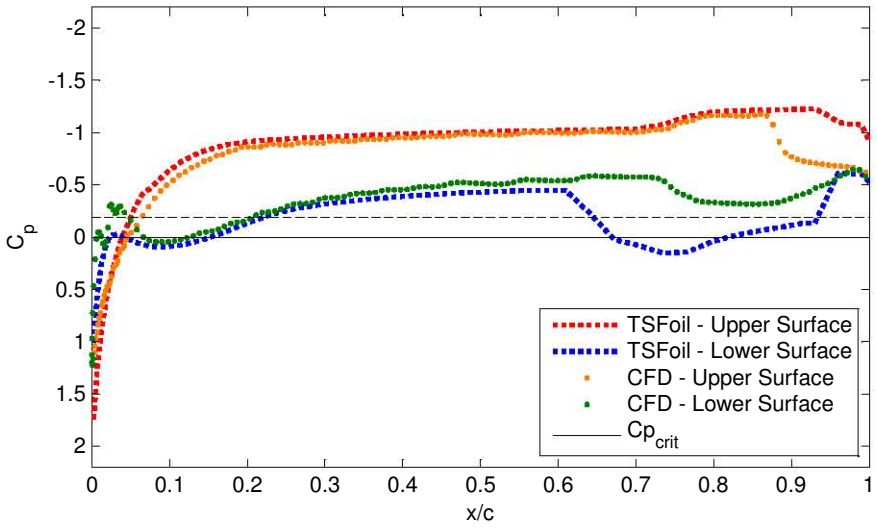


Figure 5.33: TSFoil vs CFD C_p distribution for $\alpha = 3°$, $M_\infty = 0.9$

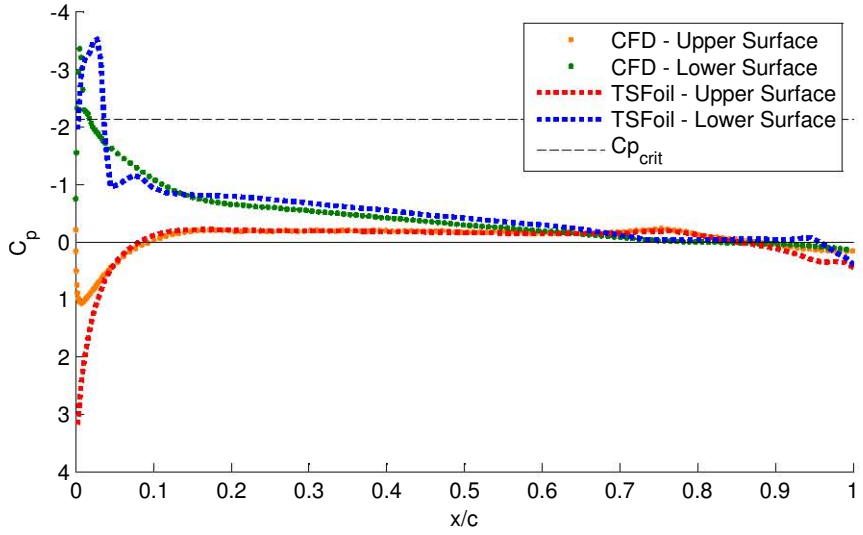


Figure 5.34: TSFoil vs CFD C_p distribution for $\alpha = -6°$, $M_\infty = 0.5$

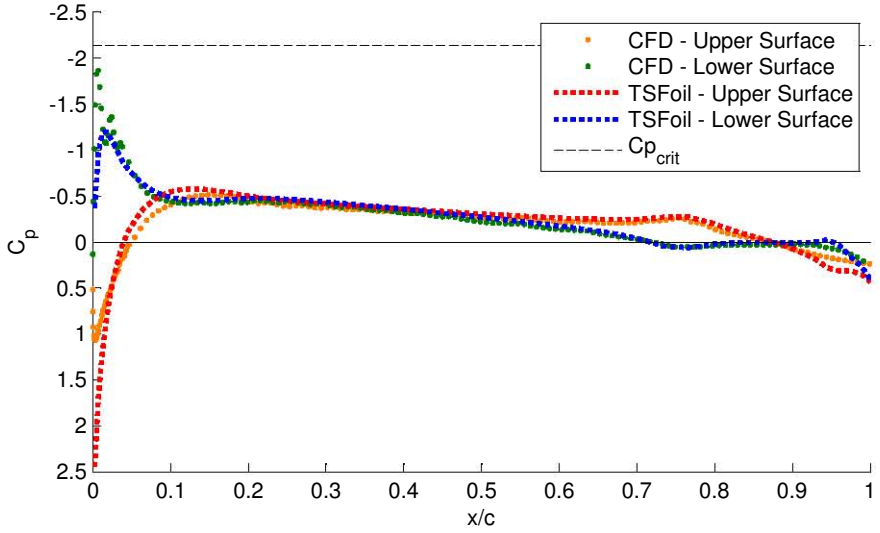


Figure 5.35: TSFoil vs CFD C_p distribution for $\alpha = -3°$, $M_\infty = 0.5$

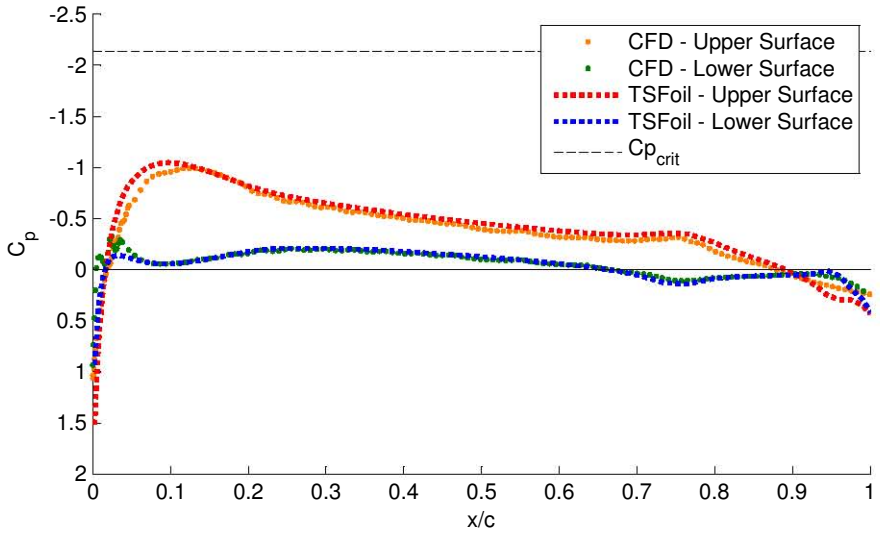


Figure 5.36: TSFoil vs CFD C_p distribution for $\alpha = 0°$, $M_\infty = 0.5$

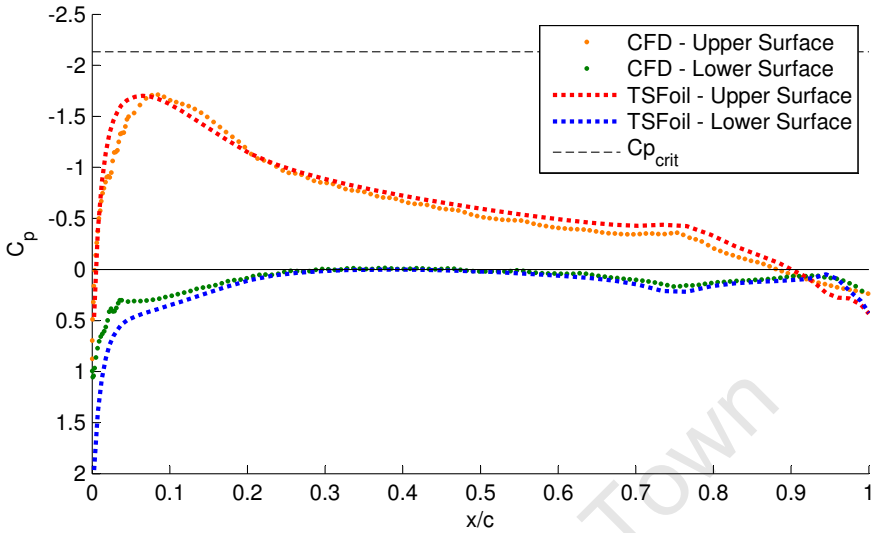


Figure 5.37: TSFoil vs CFD C_p distribution for $\alpha = 3°$, $M_\infty = 0.5$

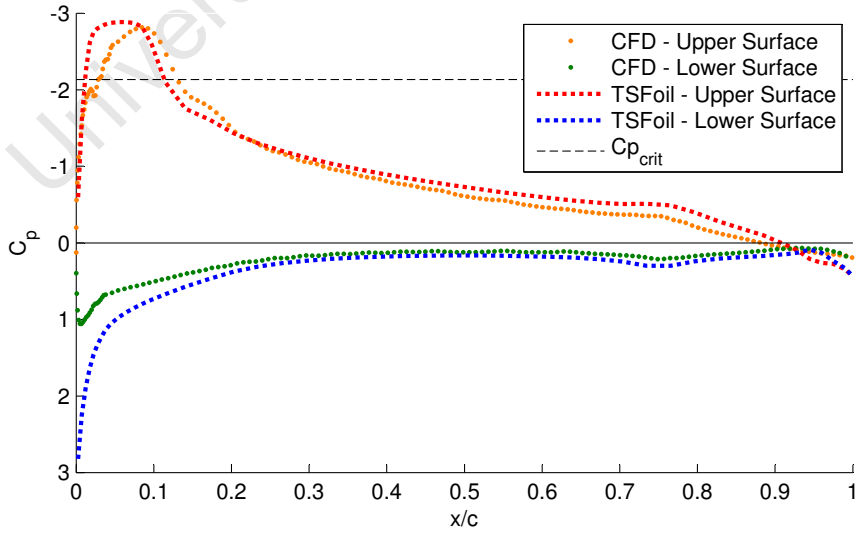


Figure 5.38: TSFoil vs CFD C_p distribution for $\alpha = 6°$, $M_\infty = 0.5$

5.8 Discussion Of Results

Application of the Kármán-Tsien Compressibility Correction

The compressibility correction, as seen in Fig. 5.1, tends to increase the value of $|C_p|$ at all points on the airfoil compared with the uncorrected inviscid flow. The difference is more pronounced the greater the value of $|C_p|$, and also for higher Mach numbers.

Viscous and Compressibility-Corrected Inviscid Linear Potential Flow

The addition of viscosity in the boundary layer to otherwise linear, inviscid, compressibility-corrected flow is marked by a slight decrease in $|C_p|$. This is caused by the change in effective airfoil profile shape due to δ^*, which leads to a decrease in curvature of the airfoil.

For the Mach-sweep seen in Figs. 5.2 to 5.7, the flow near the control surface hinges is smoothed somewhat at the lowest M_∞, but starts to approximate the inviscid solution more closely for the higher M_∞ runs, especially on the lower surface.

The irregularity near the leading edge of the upper surface is due to a laminar separation bubble, which tends to move forward on the airfoil for higher M_∞.

For the α-sweep of Figs. 5.8 to 5.13 the trend of lower $|C_p|$ in viscous flows is similar to that seen for the Mach-sweep. However, for $\alpha = 0°$ the two solutions are very closely matched everywhere except near the trailing edge. For high angles of attack ($\alpha = 10°$ and $\alpha = 15°$) moderate separation starts to occur over the upper surface near the trailing edge. The reported C_p here differs considerably from the inviscid result. XFoil claims accurate results all the way to near-stall conditions, so these results can be considered to be a fairly accurate representation of the behaviour of real flow.

Non-Linear and Compressibility-Corrected Linear Inviscid Potential Flow

TSFoil's results can be divided into subsonic and transonic regimes.

The purely subsonic results match closely with XFoil's inviscid results (Figs. 5.14 and 5.20 to 5.22). A notable difference is the smoothing of the flow near the hinges by TSFoil. TSFoil uses a much coarser grid than XFoil, where 280 panels are used in the XFoil analysis, distributed as seen in Fig. 4.4, whereas TSFoil reports only 48 grid points on each surface on the fine mesh (see §B.3). This might account for much of the smoothing effect.

For transonic cases the results can be seen to vary especially over the supersonic portion from of flow (as evidenced by $C_p > C_{p_{\text{crit}}}$). Here the drop in pressure measured by TSFoil is much more marked due to presence of the shock wave, the effects of which are effectively ignored by XFoil.

Navier-Stokes and Linear Potential Flow with Viscous Boundary Layer

The comparison of XFoil with the CFD results show good correlation at lower Mach numbers where the flow is everywhere subsonic (Figs. 5.25 and 5.26). In the CFD results, similarly to TSFoil, there is smoothing near the hinges. However this is more likely here to be an actual feature of the flow, again due to friction losses due to disturbances in the flow near these abrupt profile changes. For results which have regions of supersonic flow (Figs. 5.27 and 5.28) XFoil fails to capture the features exhibited by CFD.

Navier-Stokes and Inviscid Non-Linear Potential Flow

TSFoil shows a good correlation with the CFD results for the cases where the weak-shock assumption was not violated. This can be seen for all the α-sweep results (Figs. 5.34 to 5.38), and the low Mach number cases for the Mach-sweep (Figs. 5.29 and 5.30), with reference here to Table 5.2. However, for those cases where a violation occurred (Figs. 5.31 to 5.33), the results are wildly different, especially post-shock, and are indicative of a failure for the results to converge.

5.9 CFD Analyses

5.9.1 Introduction

Two sets of results are now presented which show more fully the true behaviour of the flow. All simulations here were again conducted on a NACA 23012 airfoil with $\delta_a = 4°$, $\delta_t = -4°$.

The first set of results fixes α at 3° while varying the freestream Mach number. Here M_∞ was varied within the range $0.5 \leq M_\infty \leq 0.9$ in steps of 0.1. These results appear in Figs. 5.39 to 5.56 on pages 73–78, and are identical in form to those explained above.

The second set of results fixes M_∞ at 0.8 and varies the angle of attack so that $\alpha = [-9, ^\circ - 6, ^\circ - 3, ^\circ 0, ^\circ 3, ^\circ 6, ^\circ 9]$. These results appear in Figs. 5.57 to 5.77 on pages 79–85. Each page features contours of three properties from a single simulation; the distribution of pressure coefficient, the distribution of Mach number, and the distribution of *effective viscosity*[1]. The Mach contours clearly show the region(s) of supersonic flow, while the viscosity contours show the extent of the wake.

The $y+$ values for the simulations undertaken are presented in Appendix D.2, at least in tabular form for all cases, with a small selection of plots which best capture the behaviour exhibited. It can be seen that in all cases the value of $y+$ falls within the range required by the Spalart-Allmaras turbulence model, as described in §3.7.1.

[1] Here the effective viscosity is practically equivalent to the turbulent viscosity, and is therefore an illustration of the turbulence in the flow.

5.9.2 Contour Plots for Mach-Sweep Simulations

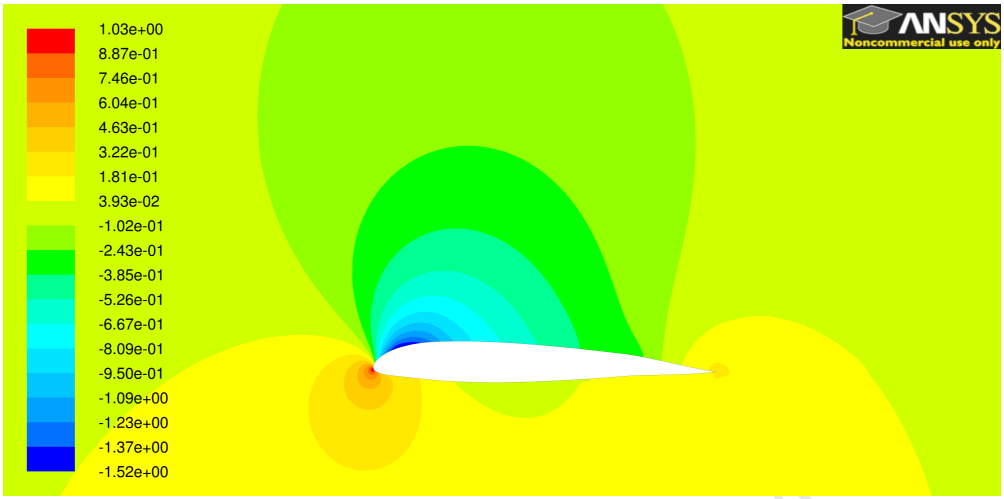


Figure 5.39: Distribution of C_p over NACA 23012 at $\alpha = 3°$, $M_\infty = 0.4$

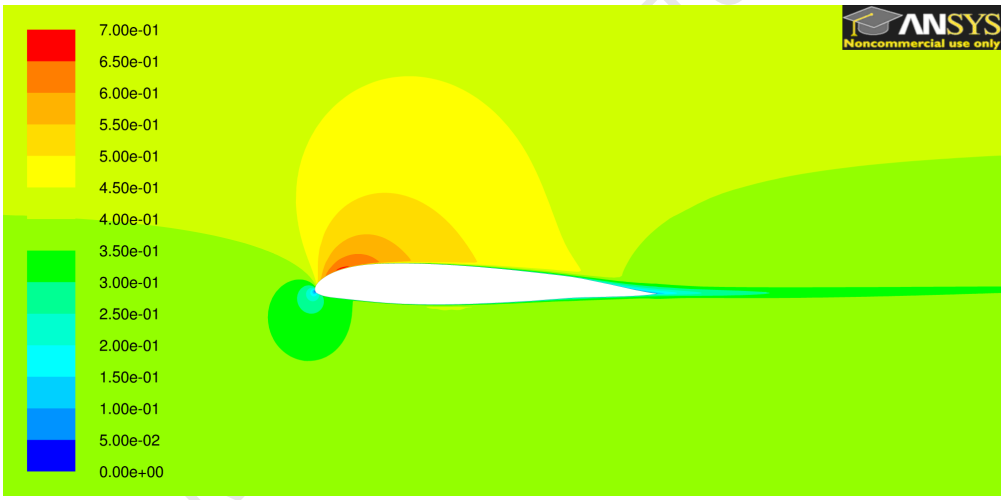


Figure 5.40: Distribution of Mach number over NACA 23012 at $\alpha = 3°$, $M_\infty = 0.4$

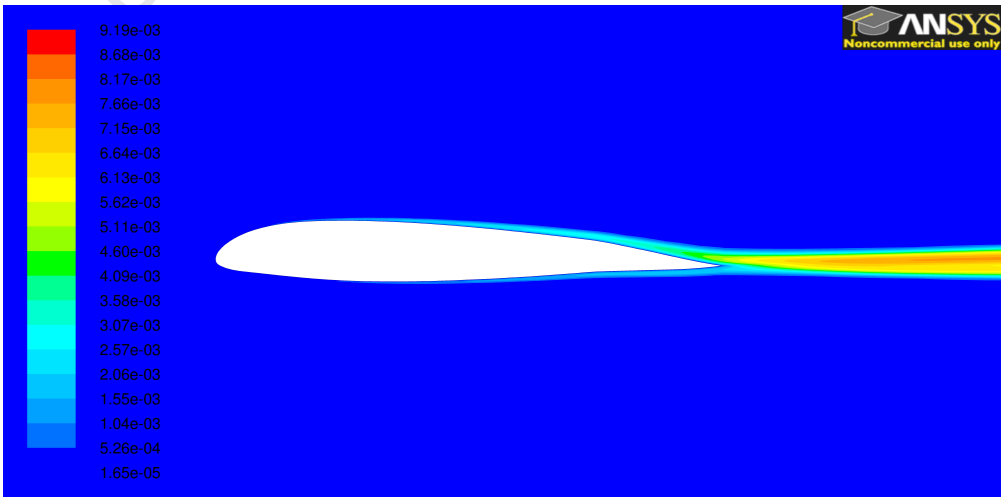


Figure 5.41: Distribution of effective viscosity over NACA 23012 at $\alpha = 3°$, $M_\infty = 0.4$

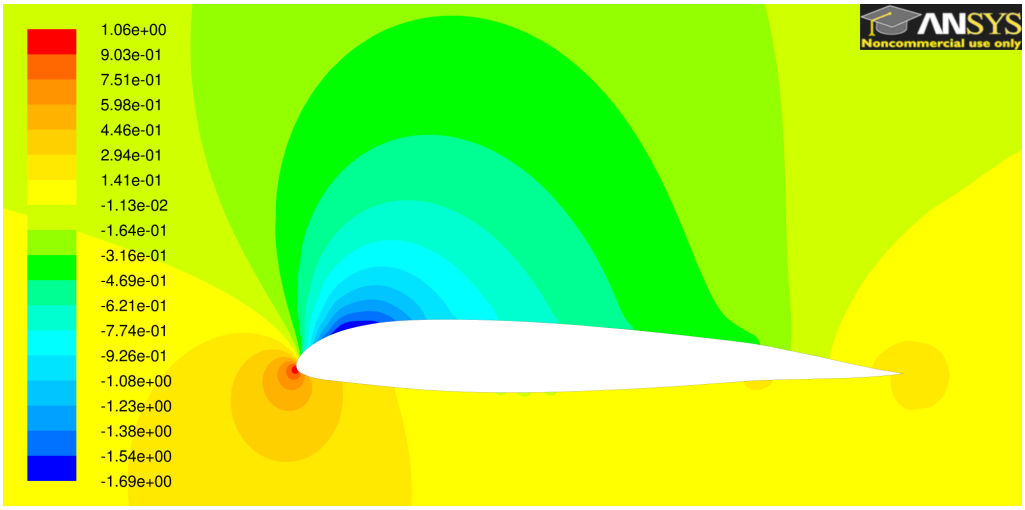


Figure 5.42: Distribution of C_p over NACA 23012 at $\alpha = 3°$, $M_\infty = 0.5$

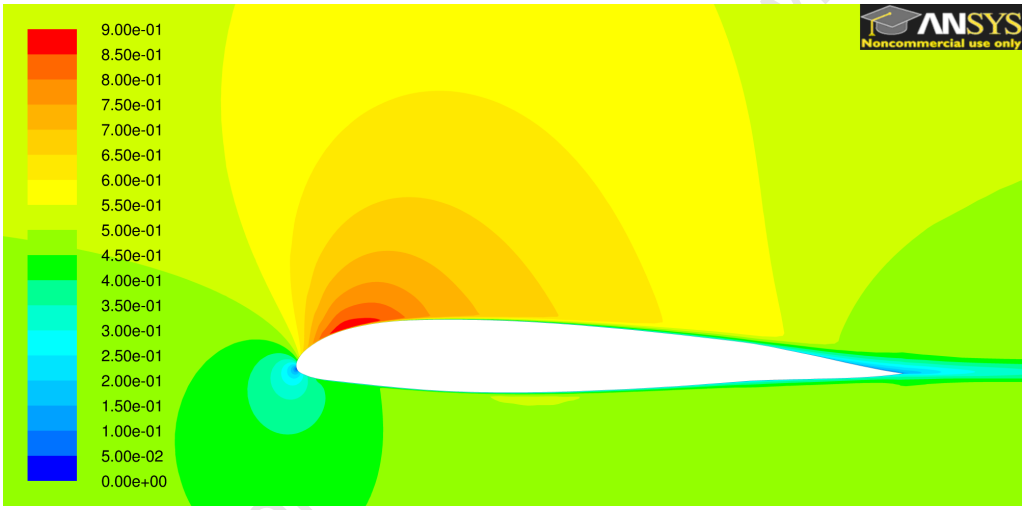


Figure 5.43: Distribution of Mach number over NACA 23012 at $\alpha = 3°$, $M_\infty = 0.5$

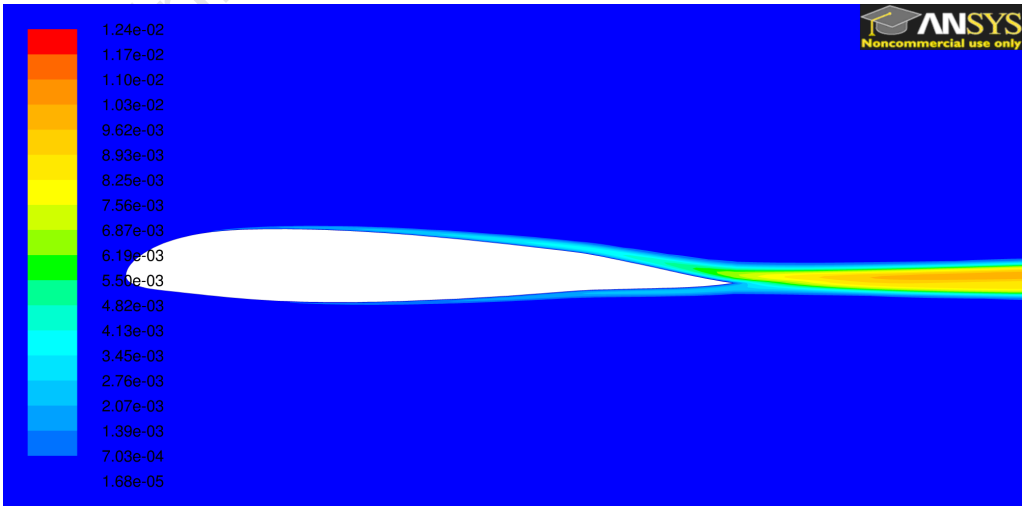


Figure 5.44: Distribution of effective viscosity over NACA 23012 at $\alpha = 3°$, $M_\infty = 0.5$

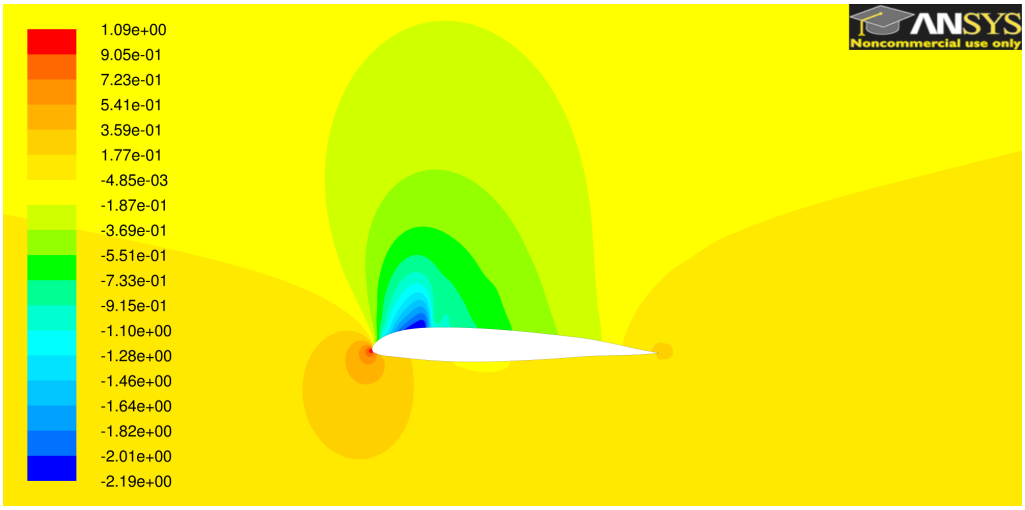


Figure 5.45: Distribution of C_p over NACA 23012 at $\alpha = 3°$, $M_\infty = 0.6$

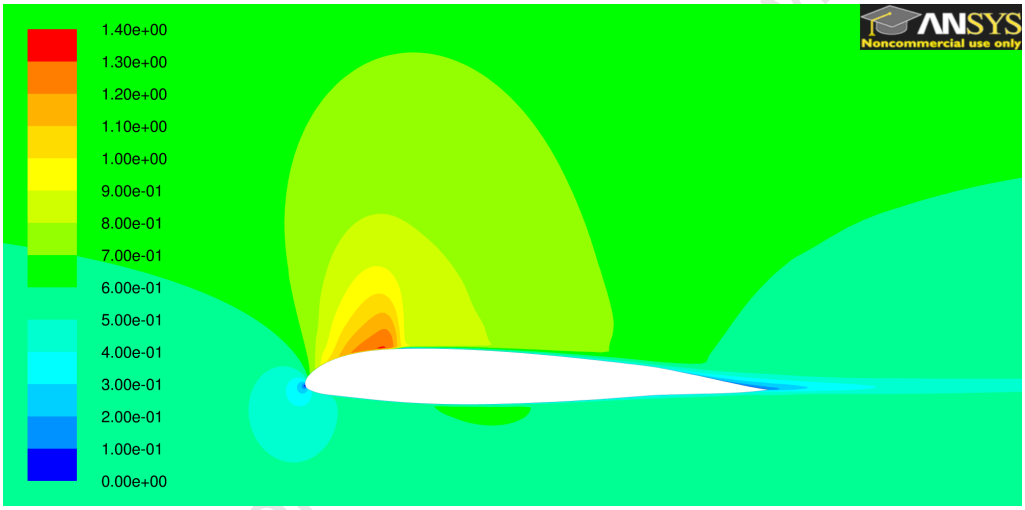


Figure 5.46: Distribution of Mach number over NACA 23012 at $\alpha = 3°$, $M_\infty = 0.6$

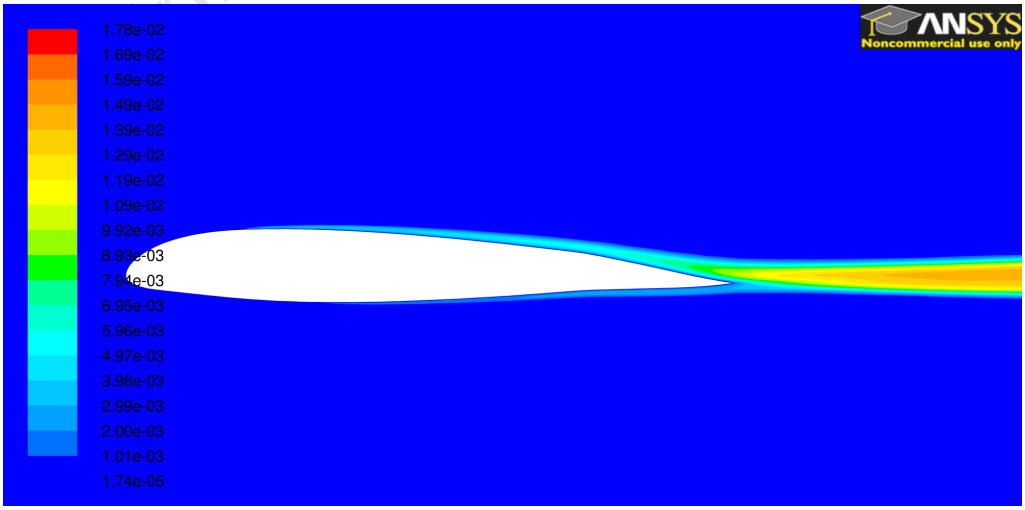


Figure 5.47: Distribution of effective viscosity over NACA 23012 at $\alpha = 3°$, $M_\infty = 0.6$

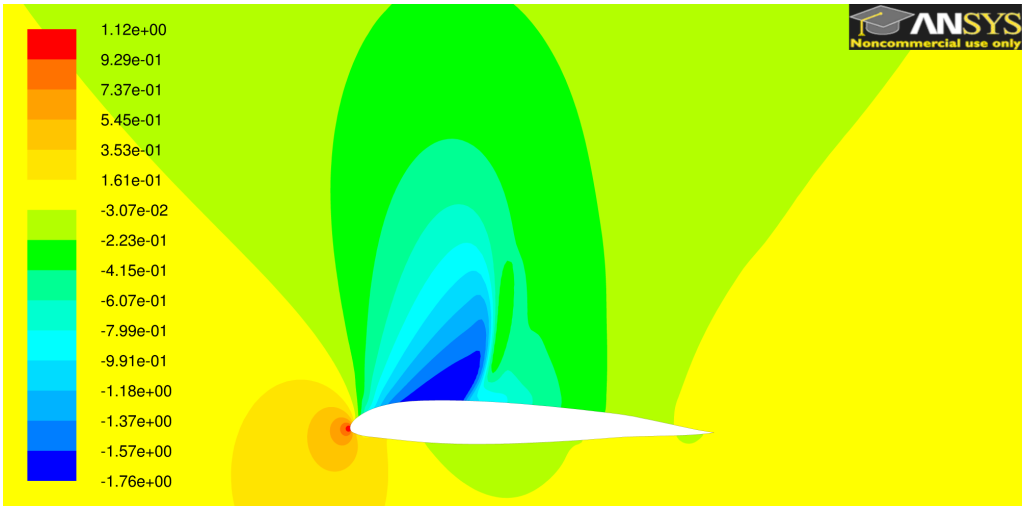


Figure 5.48: Distribution of C_p over NACA 23012 at $\alpha = 3°$, $M_\infty = 0.7$

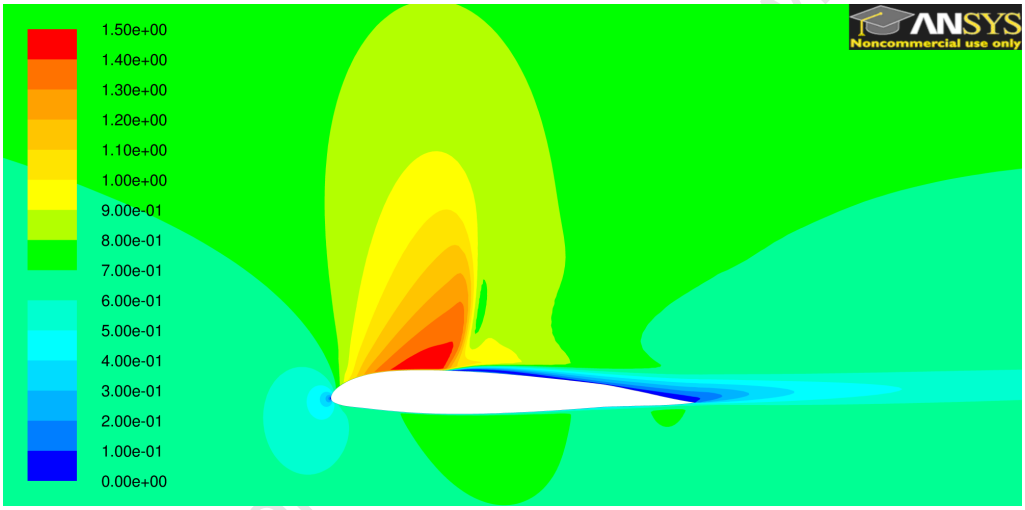


Figure 5.49: Distribution of Mach number over NACA 23012 at $\alpha = 3°$, $M_\infty = 0.7$

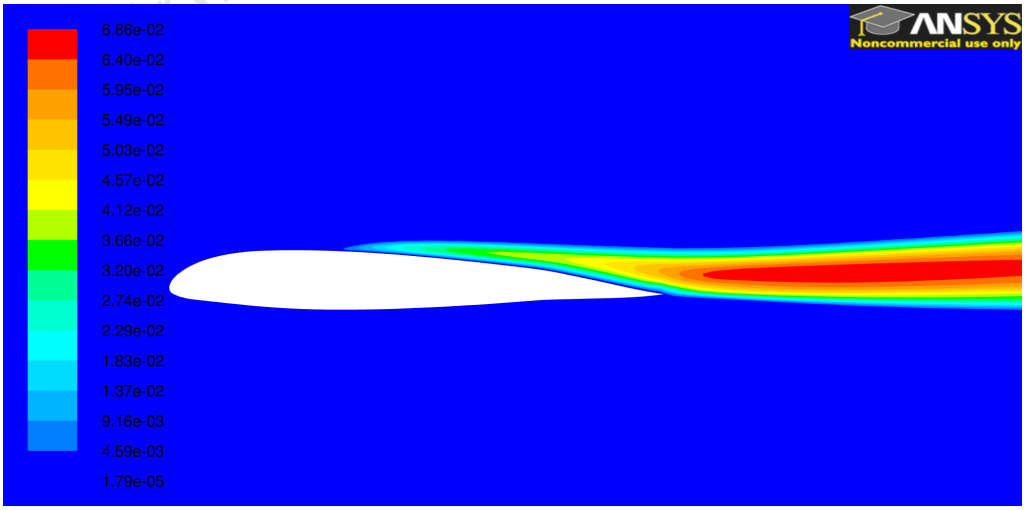


Figure 5.50: Distribution of effective viscosity over NACA 23012 at $\alpha = 3°$, $M_\infty = 0.7$

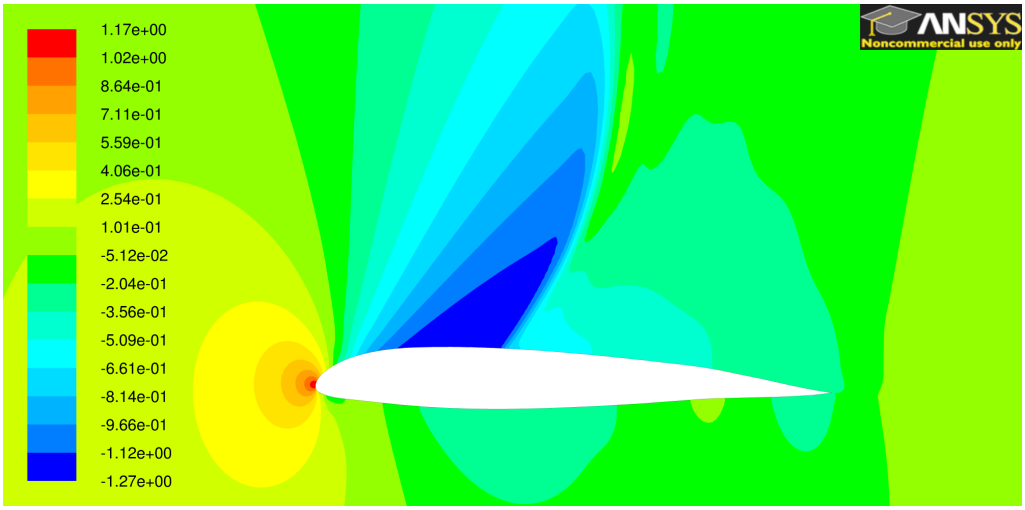


Figure 5.51: Distribution of C_p over NACA 23012 at $\alpha = 3°$, $M_\infty = 0.8$

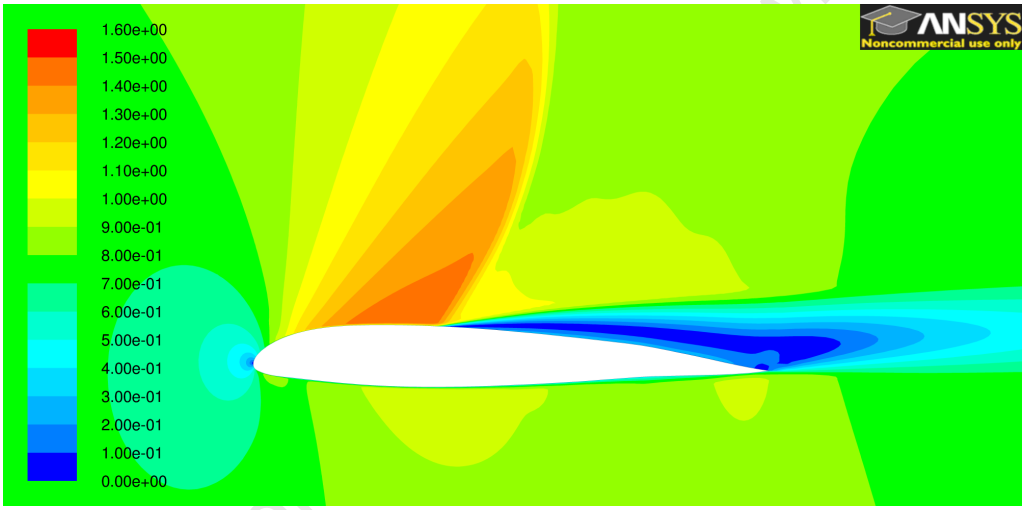


Figure 5.52: Distribution of Mach number over NACA 23012 at $\alpha = 3°$, $M_\infty = 0.8$

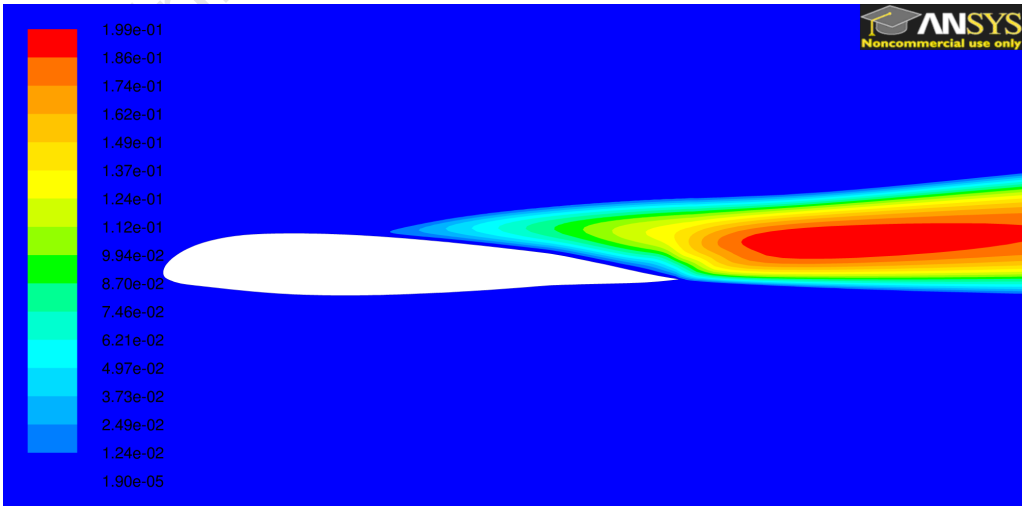


Figure 5.53: Distribution of effective viscosity over NACA 23012 at $\alpha = 3°$, $M_\infty = 0.8$

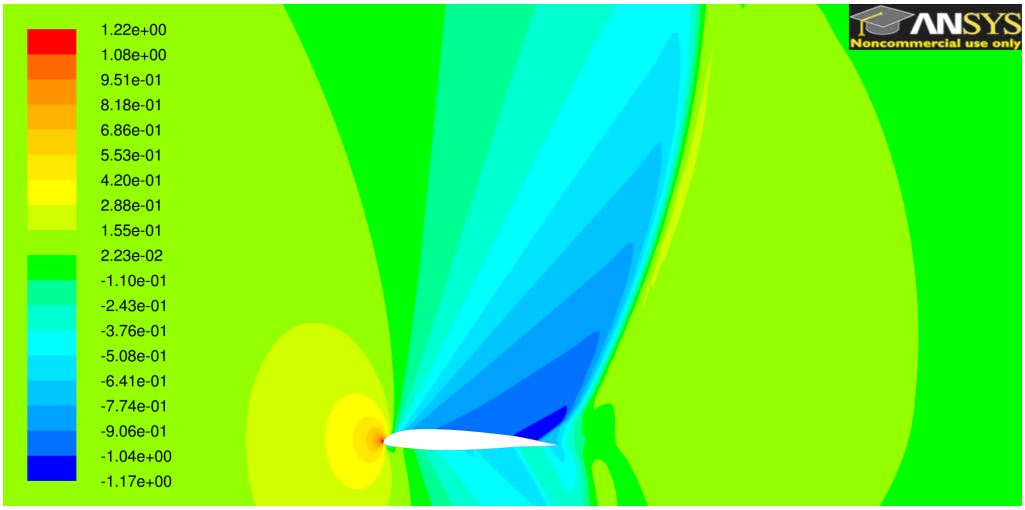


Figure 5.54: Distribution of C_p over NACA 23012 at $\alpha = 3°$, $M_\infty = 0.9$

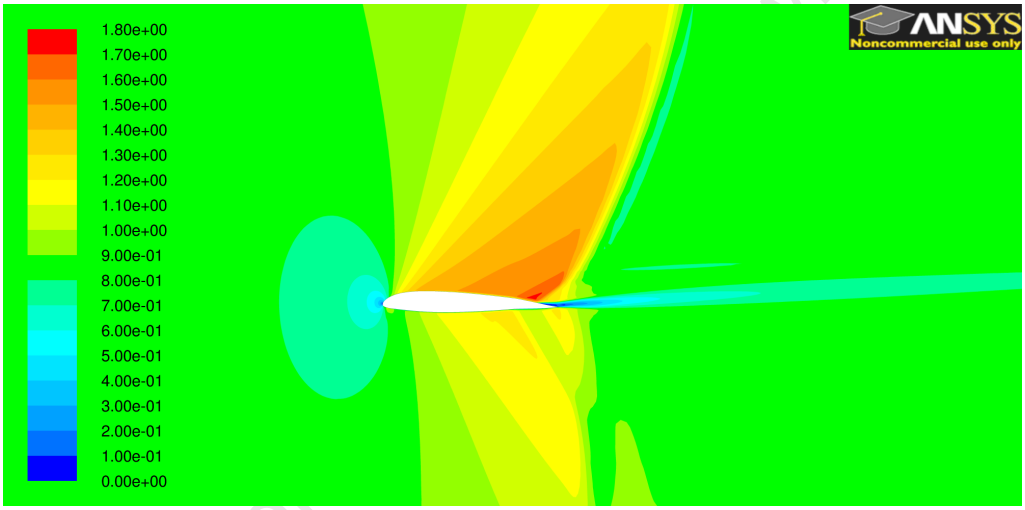


Figure 5.55: Distribution of Mach number over NACA 23012 at $\alpha = 3°$, $M_\infty = 0.9$

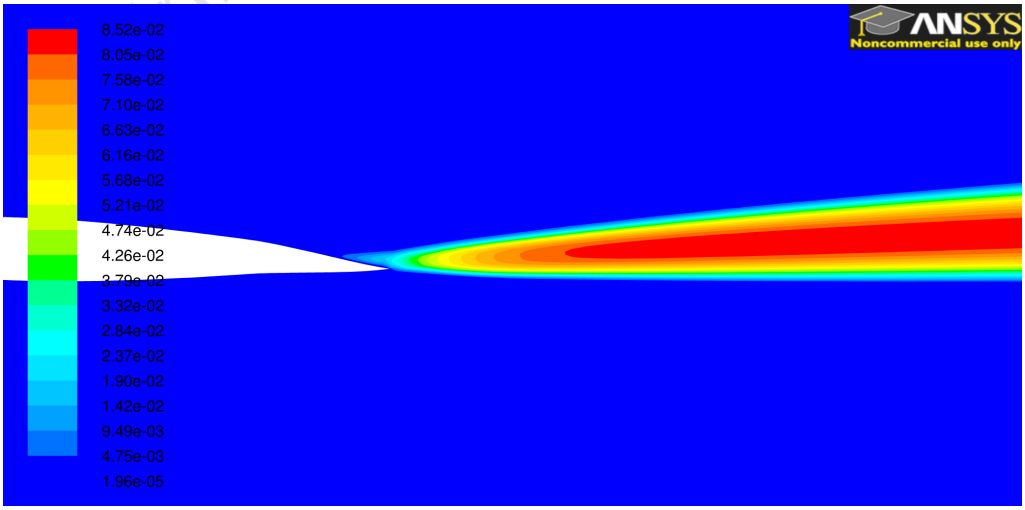


Figure 5.56: Distribution of effective viscosity over NACA 23012 at $\alpha = 3°$, $M_\infty = 0.9$

5.9.3 Contour Plots for α-Sweep Simulations

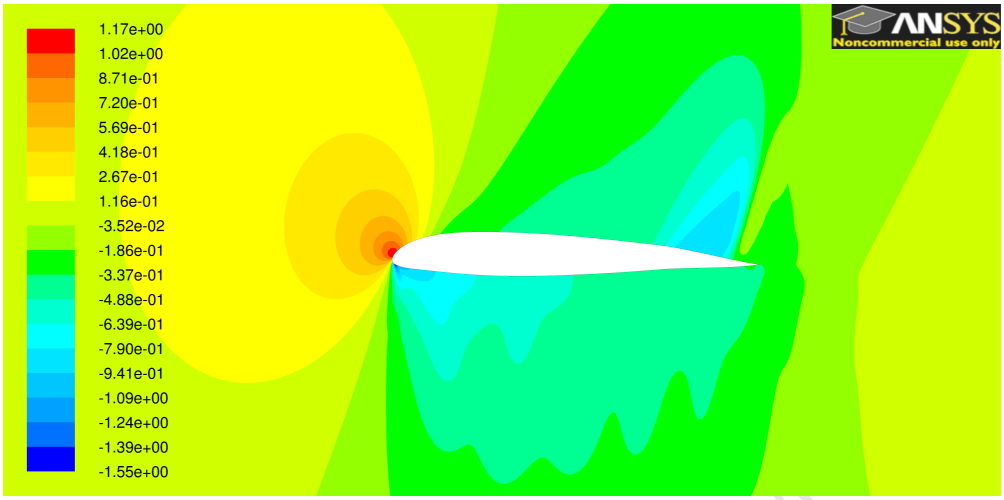


Figure 5.57: Distribution of C_p over NACA 23012 at $M_\infty = 0.8$, $\alpha = -9°$

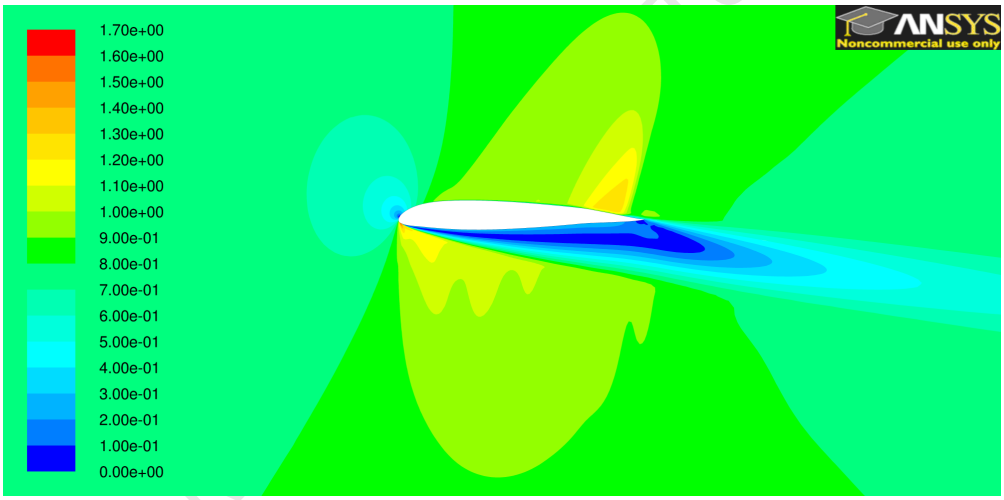


Figure 5.58: Distribution of Mach number over NACA 23012 at $M_\infty = 0.8$, $\alpha = -9°$

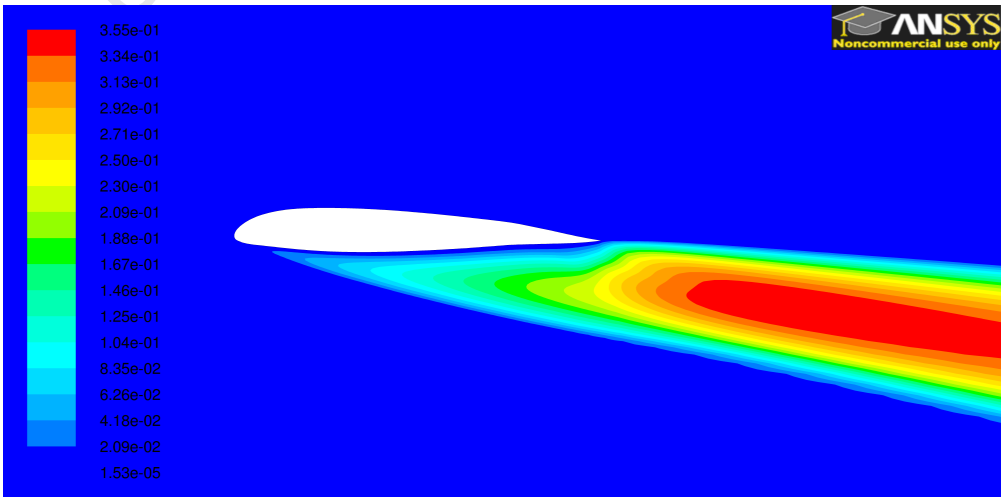


Figure 5.59: Distribution of effective viscosity over NACA 23012 at $M_\infty = 0.8$, $\alpha = -9°$

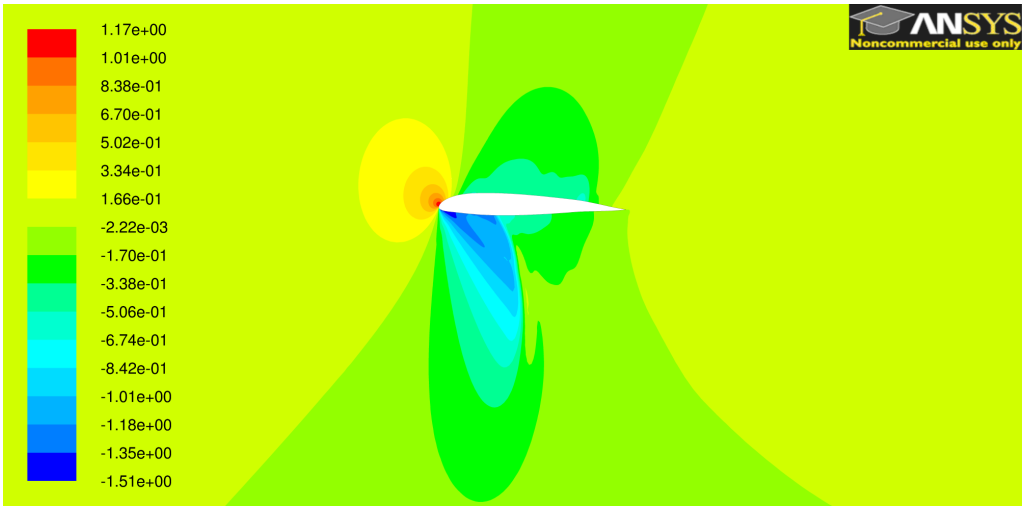


Figure 5.60: Distribution of C_p over NACA 23012 at $M_\infty = 0.8$, $\alpha = -6°$

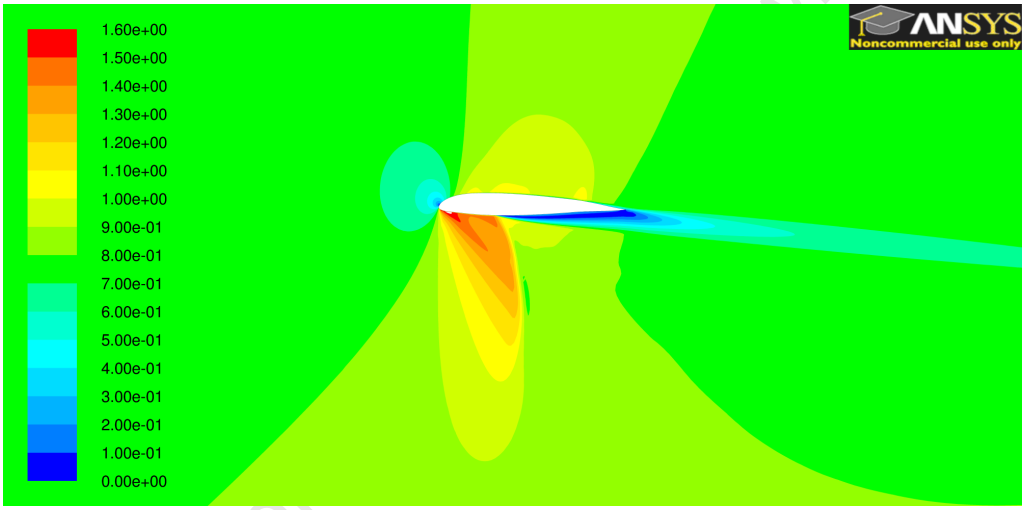


Figure 5.61: Distribution of Mach number over NACA 23012 at $M_\infty = 0.8$, $\alpha = -6°$

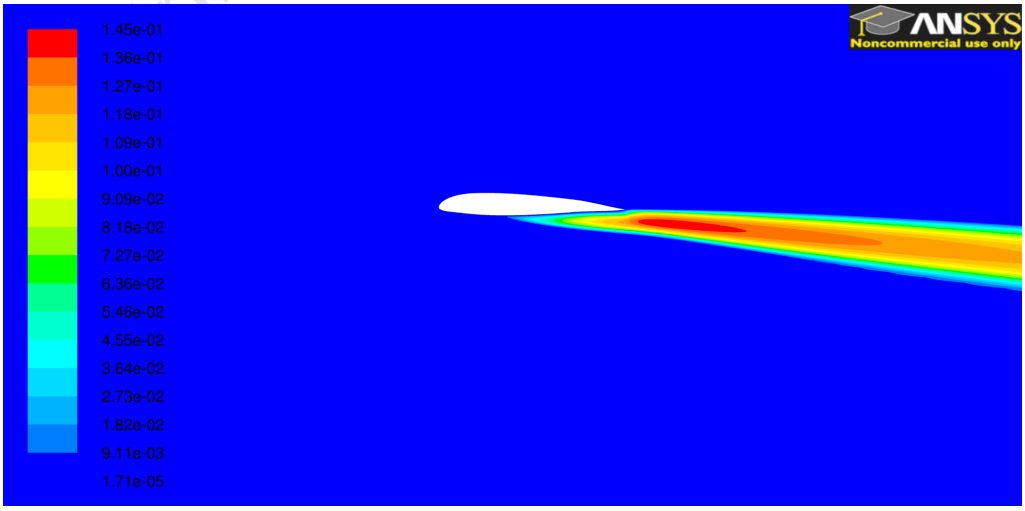


Figure 5.62: Distribution of effective viscosity over NACA 23012 at $M_\infty = 0.8$, $\alpha = -6°$

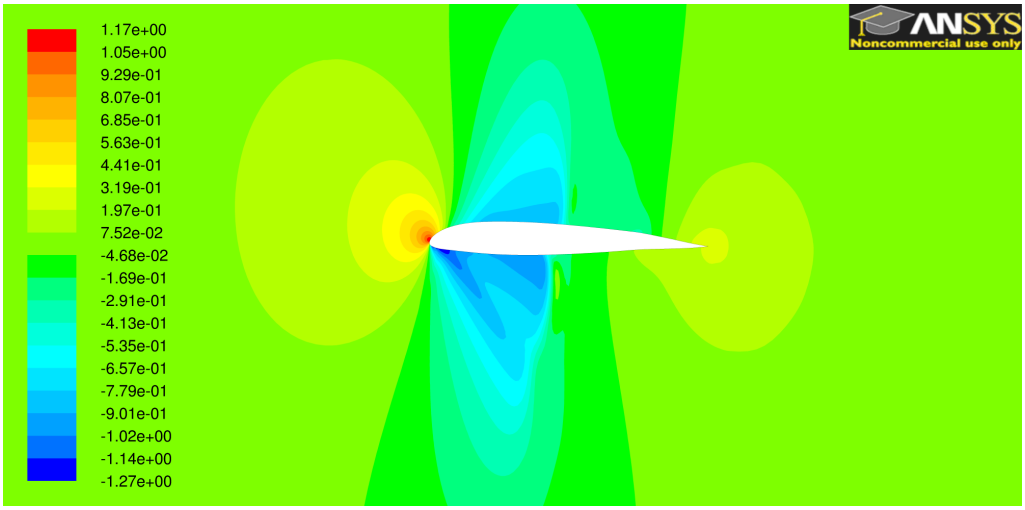


Figure 5.63: Distribution of C_p over NACA 23012 at $M_\infty = 0.8$, $\alpha = -3°$

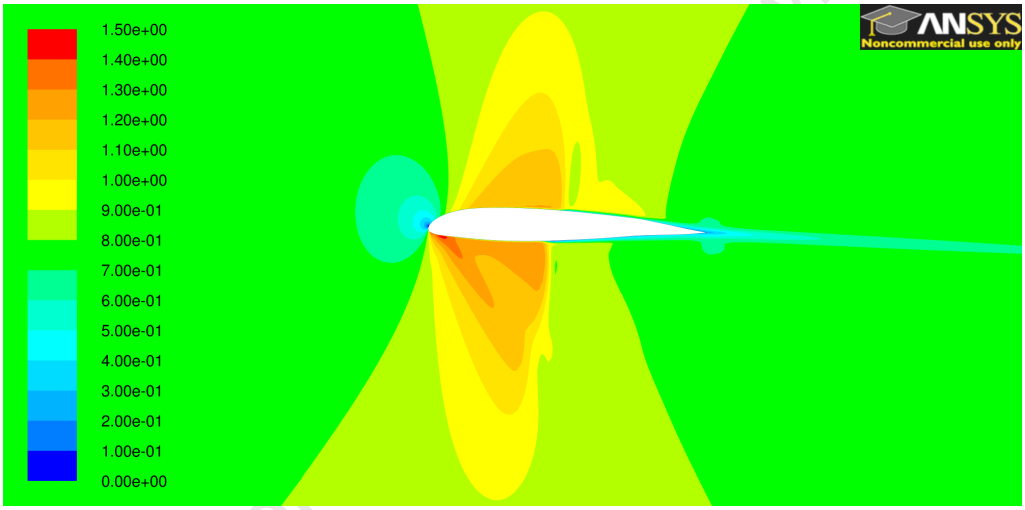


Figure 5.64: Distribution of Mach number over NACA 23012 at $M_\infty = 0.8$, $\alpha = -3°$

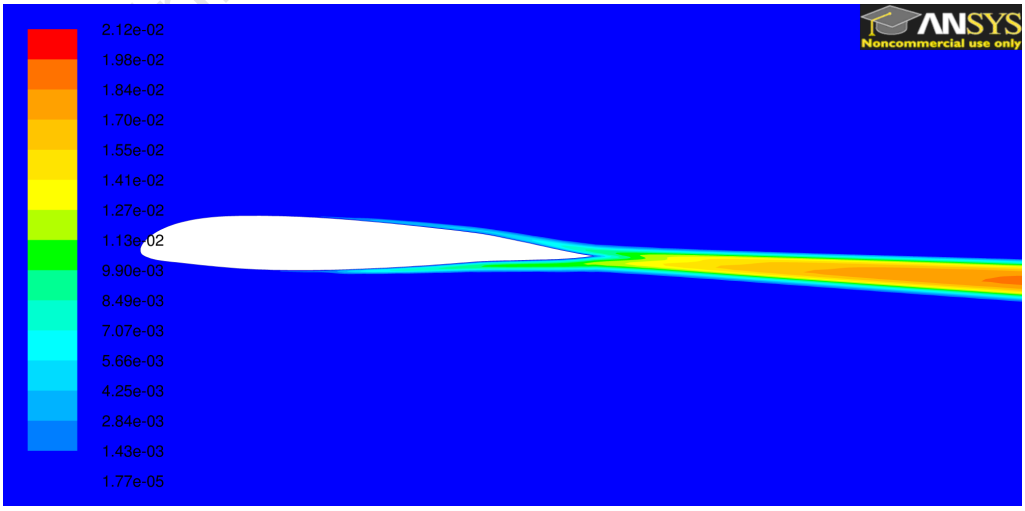


Figure 5.65: Distribution of effective viscosity over NACA 23012 at $M_\infty = 0.8$, $\alpha = -3°$

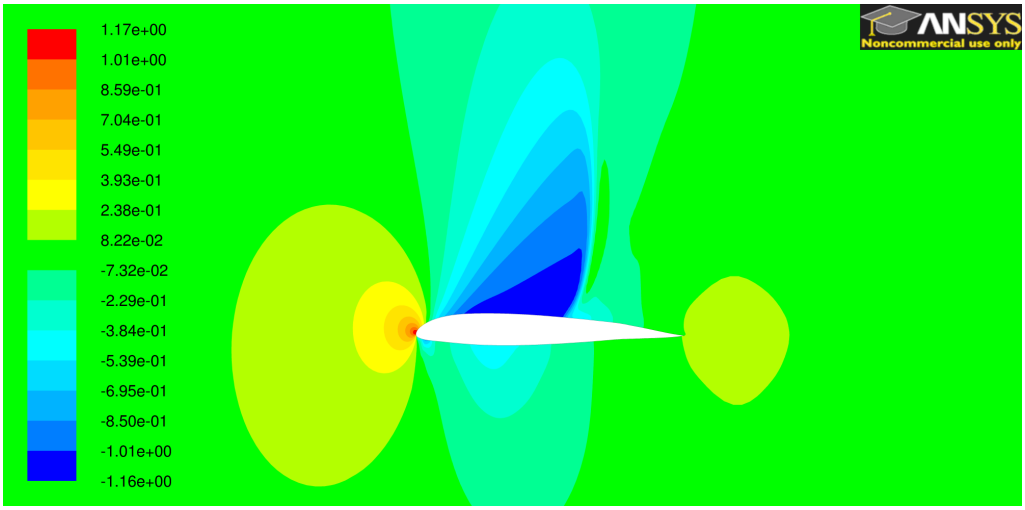


Figure 5.66: Distribution of C_p over NACA 23012 at $M_\infty = 0.8$, $\alpha = 0°$

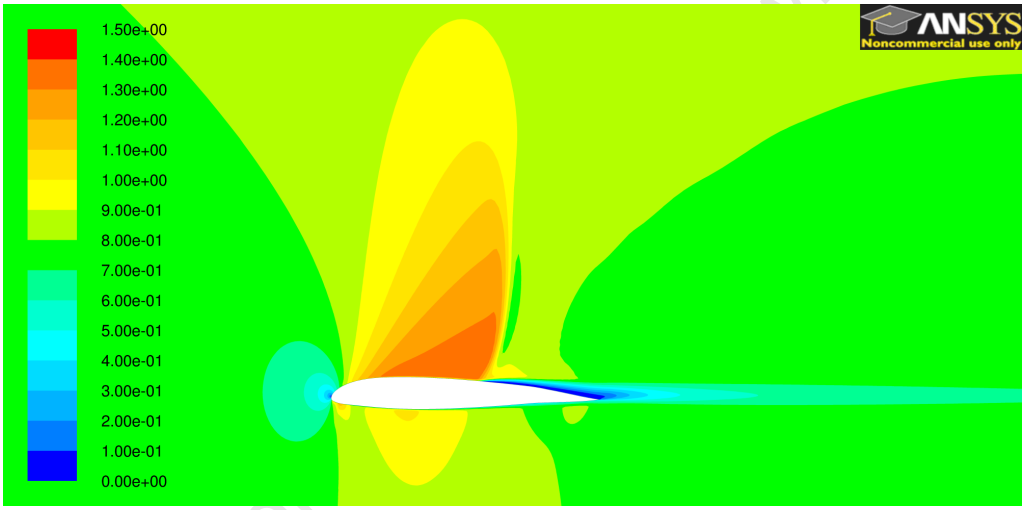


Figure 5.67: Distribution of Mach number over NACA 23012 at $M_\infty = 0.8$, $\alpha = 0°$

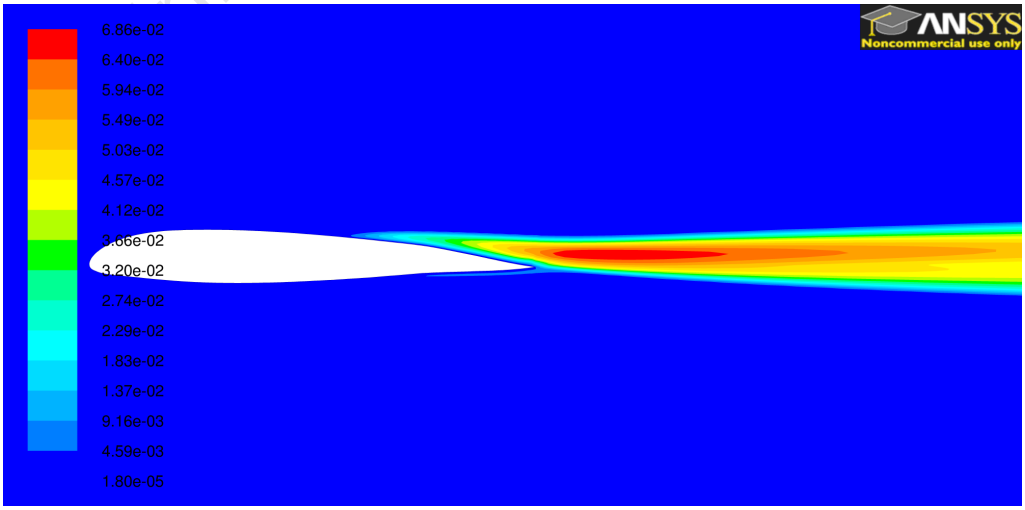


Figure 5.68: Distribution of effective viscosity over NACA 23012 at $M_\infty = 0.8$, $\alpha = 0°$

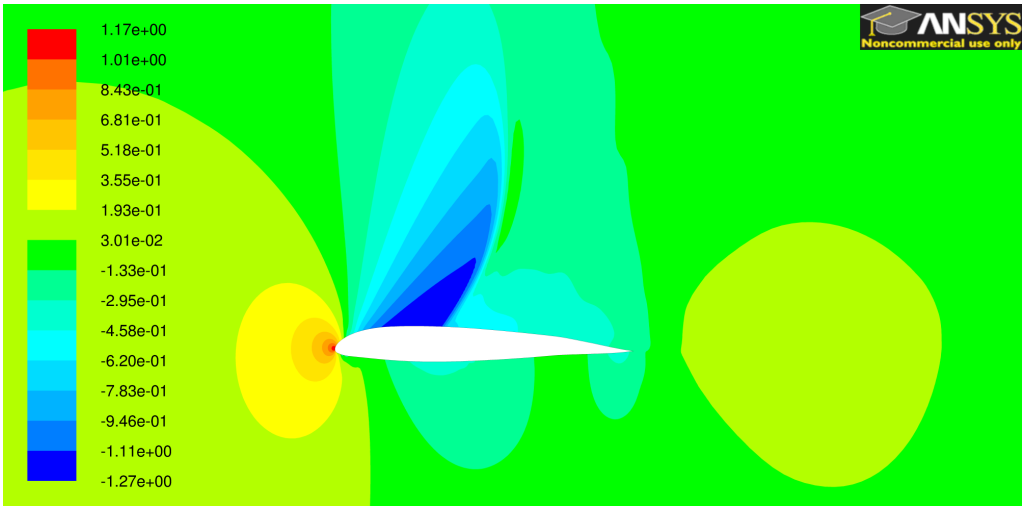


Figure 5.69: Distribution of C_p over NACA 23012 at $M_\infty = 0.8$, $\alpha = 3°$

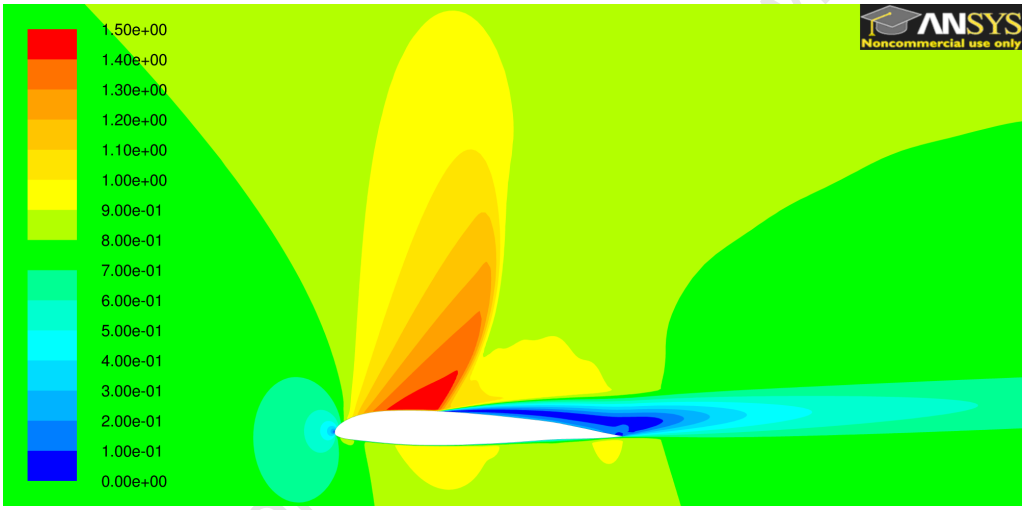


Figure 5.70: Distribution of Mach number over NACA 23012 at $M_\infty = 0.8$, $\alpha = 3°$

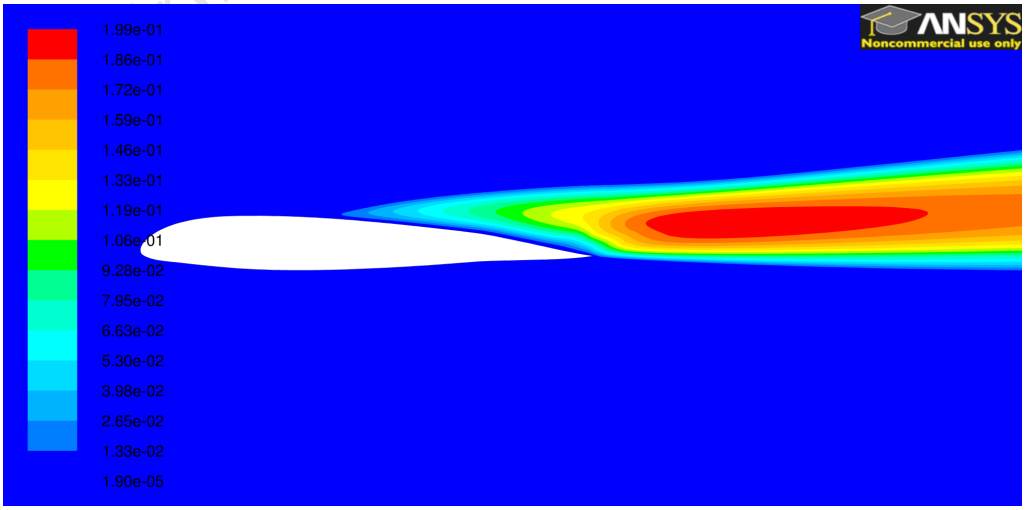


Figure 5.71: Distribution of effective viscosity over NACA 23012 at $M_\infty = 0.8$, $\alpha = 3°$

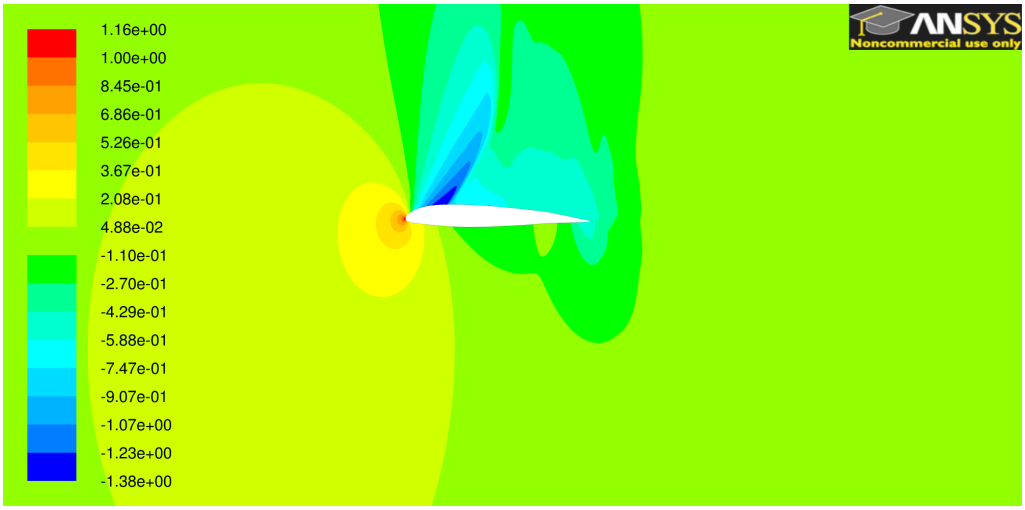


Figure 5.72: Distribution of C_p over NACA 23012 at $M_\infty = 0.8$, $\alpha = 6°$

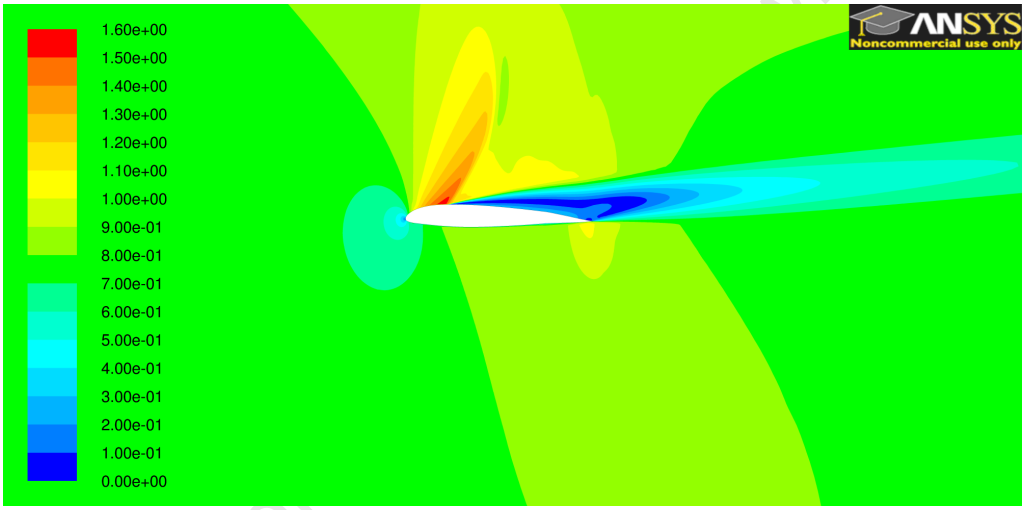


Figure 5.73: Distribution of Mach number over NACA 23012 at $M_\infty = 0.8$, $\alpha = 6°$

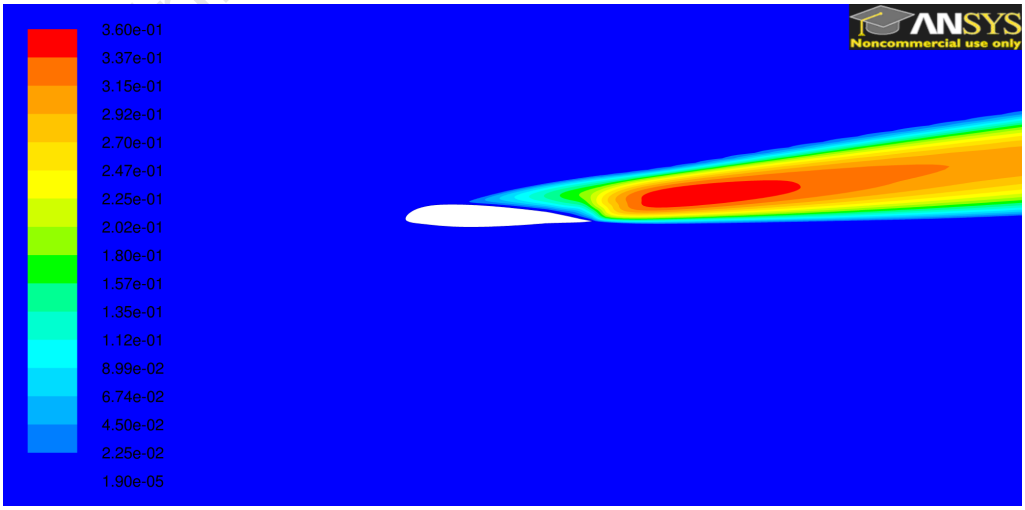


Figure 5.74: Distribution of effective viscosity over NACA 23012 at $M_\infty = 0.8$, $\alpha = 6°$

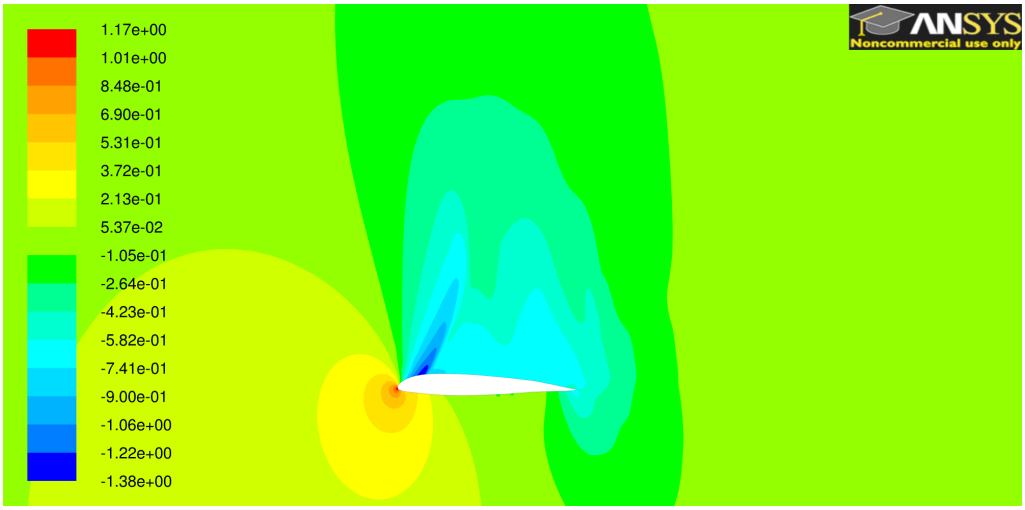


Figure 5.75: Distribution of C_p over NACA 23012 at $M_\infty = 0.8$, $\alpha = 9°$

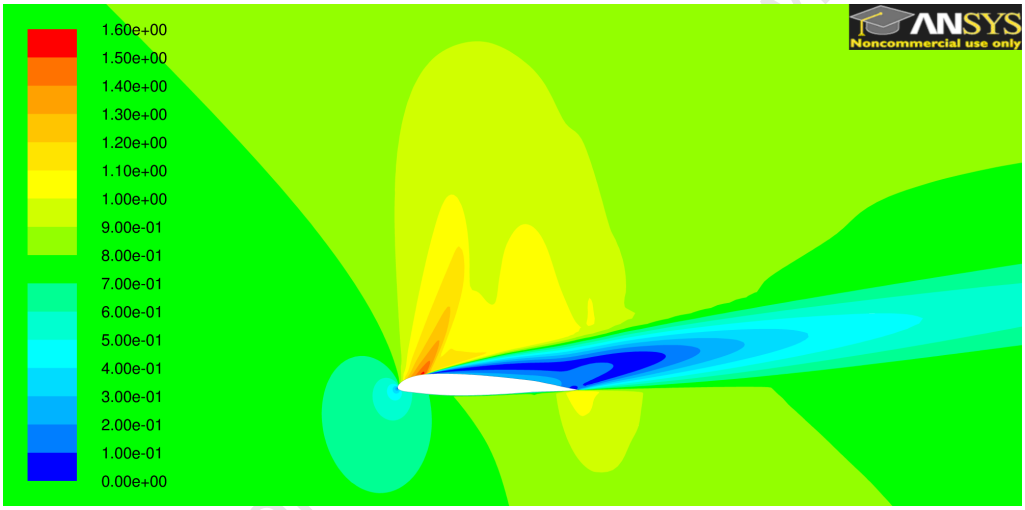


Figure 5.76: Distribution of Mach number over NACA 23012 at $M_\infty = 0.8$, $\alpha = 9°$

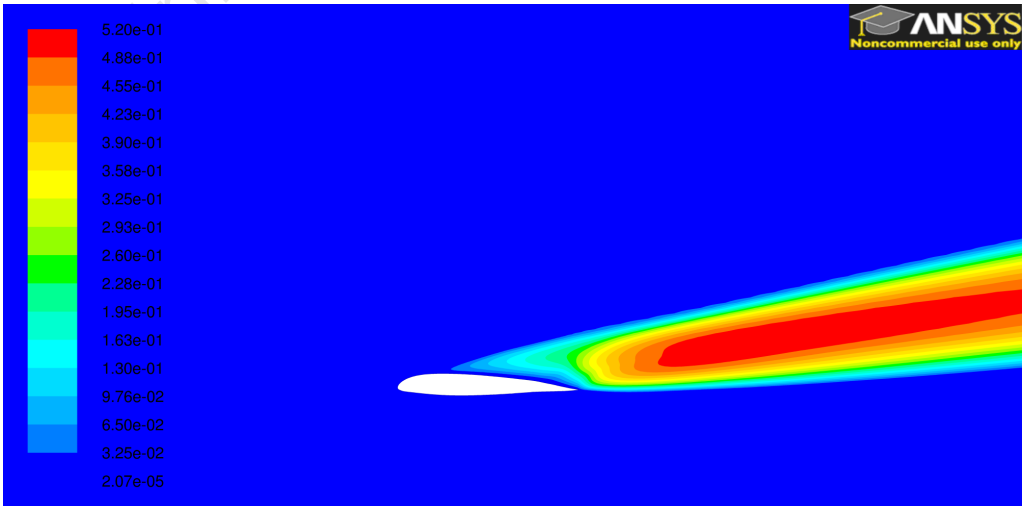


Figure 5.77: Distribution of effective viscosity over NACA 23012 at $M_\infty = 0.8$, $\alpha = 9°$

5.9.4 A Closer Look At Shock Capturing

In this section the shock-capturing capability of CFD is more closely scrutinised. Shock waves give rise to discontinuities in the flow solution, where abrupt changes in the value of flow properties occur over the mean free path of the air, which is much smaller than a single mesh interval. The ability of the CFD method used to capture this behaviour will be examined.

Figure 5.78 shows how fluid flow properties *should* change over a normal shock wave, while Figs. 5.66 and 5.67 on page 82, and Figs. 5.80 to 5.82 starting overleaf, show outputs from the CFD solver of C_p, M, ρ, T, s and h respectively.

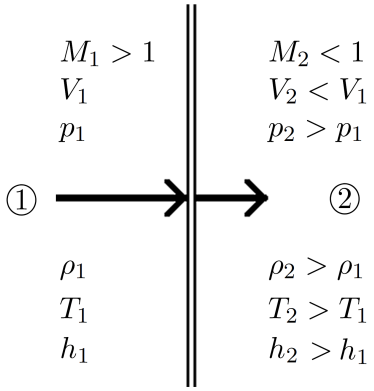


Figure 5.78: Qualitative characteristics of flow through a normal shock wave (adapted from [2])

All the contours show a similar pattern, where the property within the region of supersonic flow is either much greater or lower than the freestream value. Shown in Fig. 5.79 is a rough qualitative depiction of a generic property that increases in the supersonic region of flow then drops over the shock. Properties either display this pattern or they mirror it about the x-axis. Here precise values were not retrieved from the solution, but the contours give an indication of these trends.

This illustrates that CFD cannot exactly capture the discontinuous behaviour of the flow over a shock wave. However, the trends over the shock have been correctly predicted.

The trends from the CFD output are noted in Table 5.3 for each of the properties given in Fig. 5.78 to ensure that the qualitative characteristics are captured correctly. No exceptions were noted.

Table 5.3: Trends for the properties given in Fig. 5.78

Property	Variable	Figure	Page	Trend over shock
Pressure coefficient	C_p	5.66	82	$C_{p_2} > C_{p_1}$
Velocity	M	5.67	82	$M_2 < M_1$
Density	ρ	5.81	88	$\rho_2 > \rho_1$
Temperature	T	5.80	87	$T_2 > T_1$
Enthalpy	h	5.82	88	$h_2 > h_1$

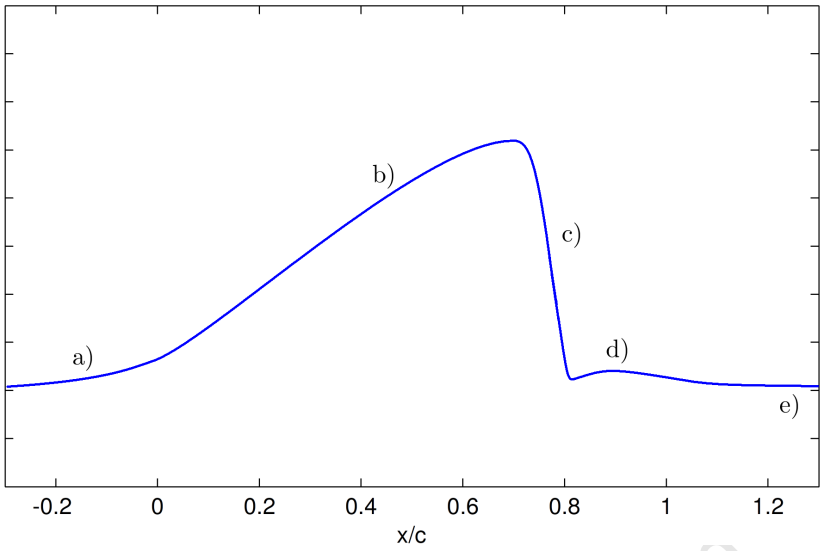


Figure 5.79: Qualitative depiction of a typical property that decreases over a shock, taken on a horizontal line above the airfoil roughly mid-shock. a) shows the property rising from its freestream value, increasing in the supersonic region b), then dropping across the shock at c), recovering slightly at d), then settling back to the freestream value at e).

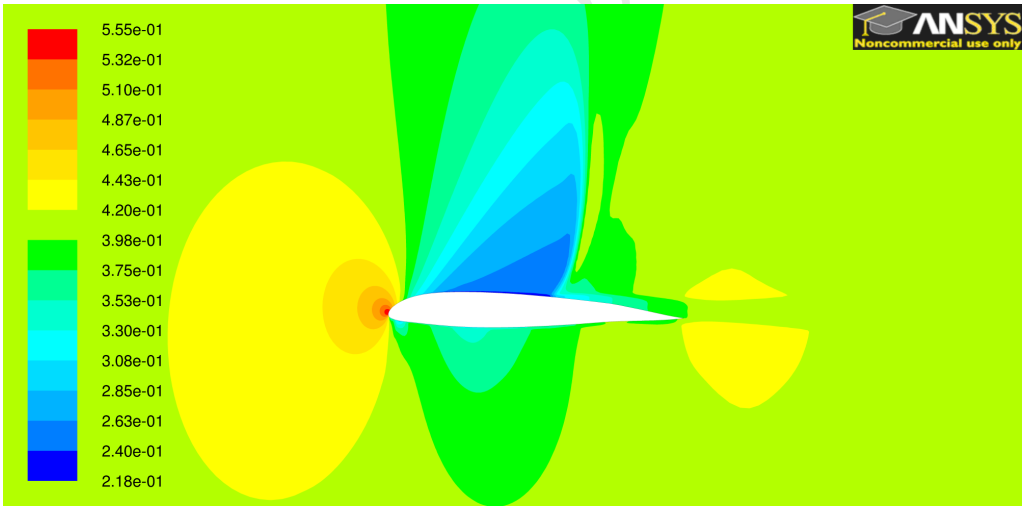


Figure 5.80: Density, ρ, in kg/m^3 for $M_\infty = 0.8$, $\alpha = 0°$

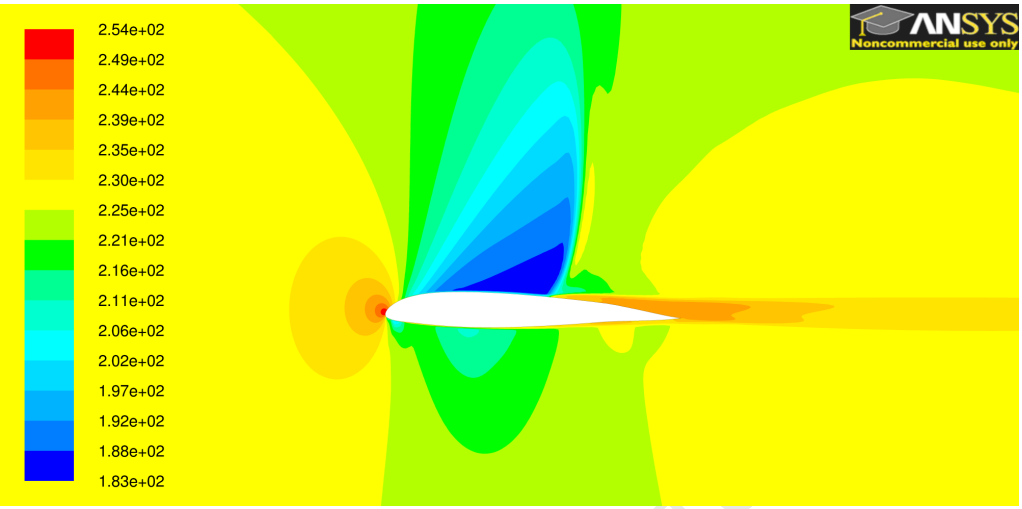


Figure 5.81: Static temperature, T, in K for $M_\infty = 0.8$, $\alpha = 0°$

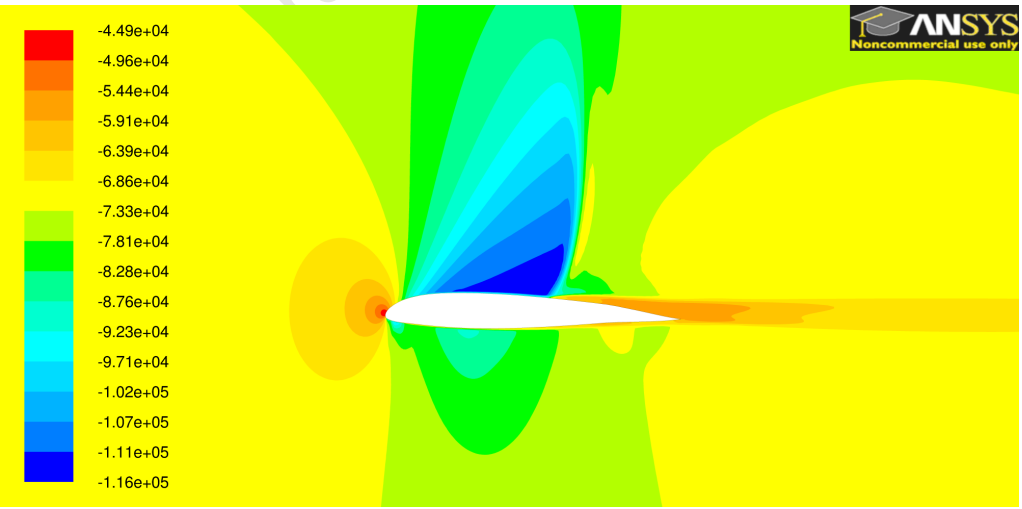


Figure 5.82: Enthalpy, h, in J/kg for $M_\infty = 0.8$, $\alpha = 0°$

5.9.5 A Closer Look At Shock/Boundary-Layer Interaction

Here we look at how the flow adjacent to the airfoil is affected by the supersonic region of flow, and by the shock itself. Attention is drawn to the effective viscosity plots from the Mach-sweep simulations, shown in Figs. 5.41 to 5.56 starting on page 73.

At freestream Mach numbers of 0.4 and 0.5 a region of turbulent flow can be seen to start very roughly at about the point of maximum thickness. Closer inspection of the boundary layer shows the flow is well-behaved (i.e. not separated). This can be seen in Fig. 5.83, with the effective viscosity shown in Fig. 5.84 for the same region of flow. A very similar velocity profile is seen extending all the way to the trailing edge, showing no signs of separation.

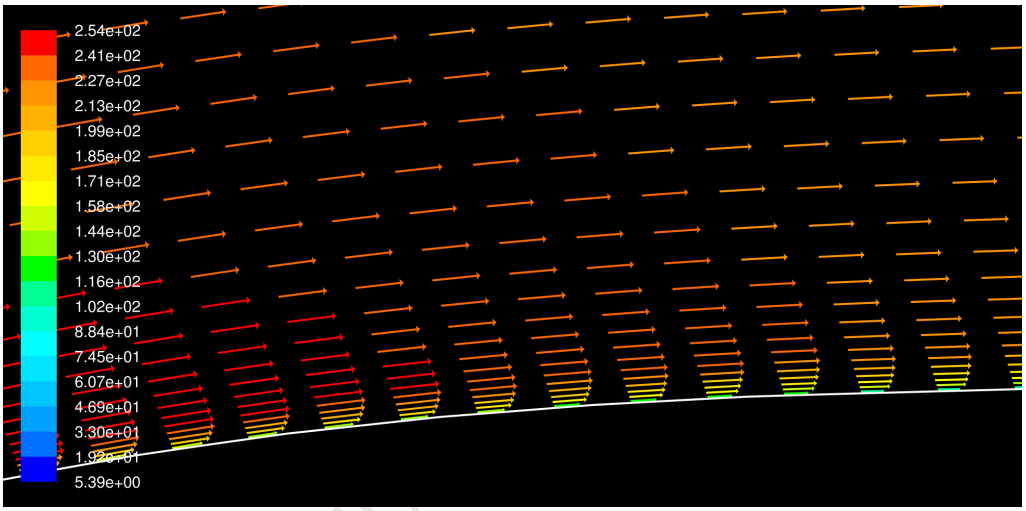


Figure 5.83: Close-up of region where an increase in viscosity is first experienced at $M_\infty = 0.5$, showing velocity vectors coloured to show regions in m/s.

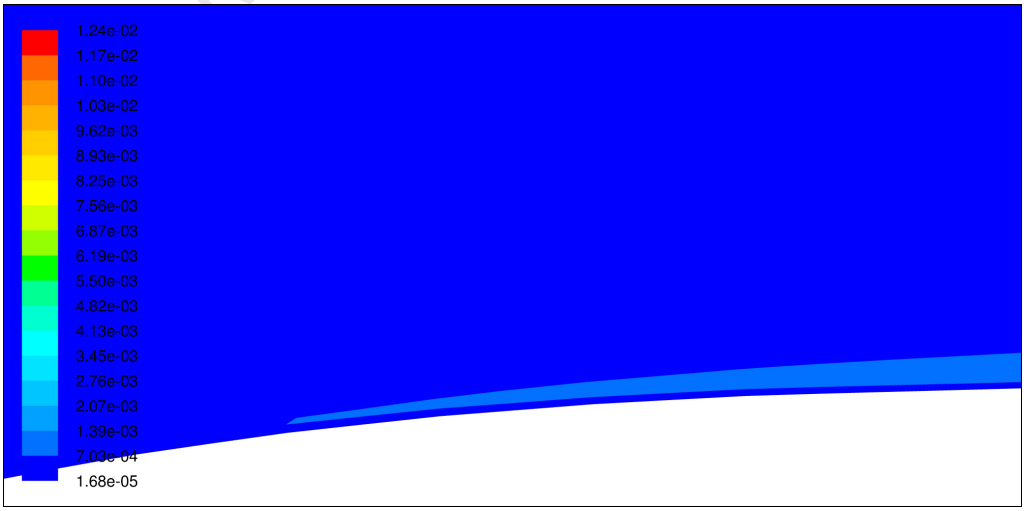


Figure 5.84: Close-up of region where turbulence is first experienced at $M_\infty = 0.5$, showing contours of effective viscosity.

We look now at the case where $M_\infty = 0.8$. Close-ups of the airfoil in the vicinity of the shock wave are shown in Figs. 5.85 and 5.86, where the same region has been shown in both plots. The contour plot of effective viscosity shows that any large-scale turbulence has been delayed until the shock, and shows a large turbulent region has formed immediately post-shock. The velocity vectors indicate the mean velocity has slowed considerably. Closer examination of the flow in Fig. 5.87 shows that separation has been initiated at the base of the shock.

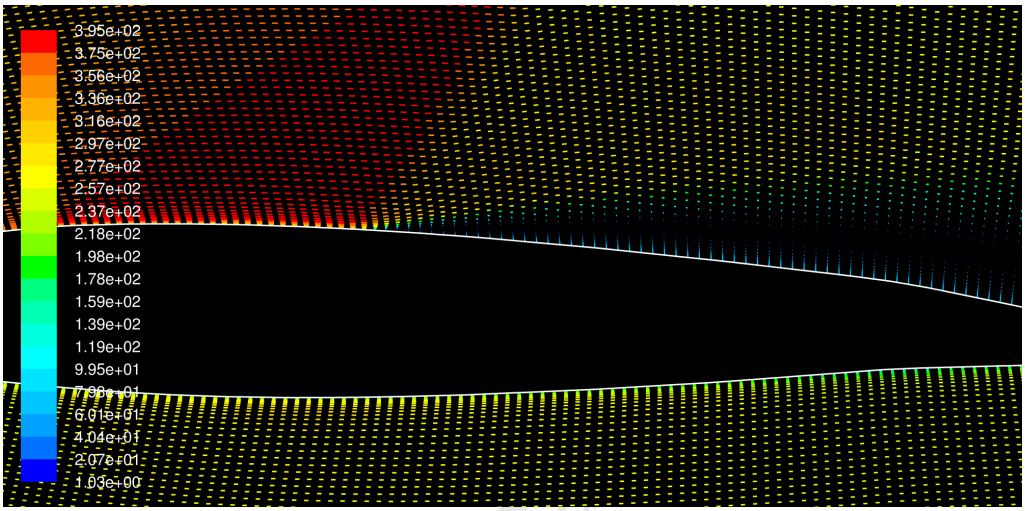


Figure 5.85: Close-up of region where turbulence is first experienced at $M_\infty = 0.8$, showing velocity vectors

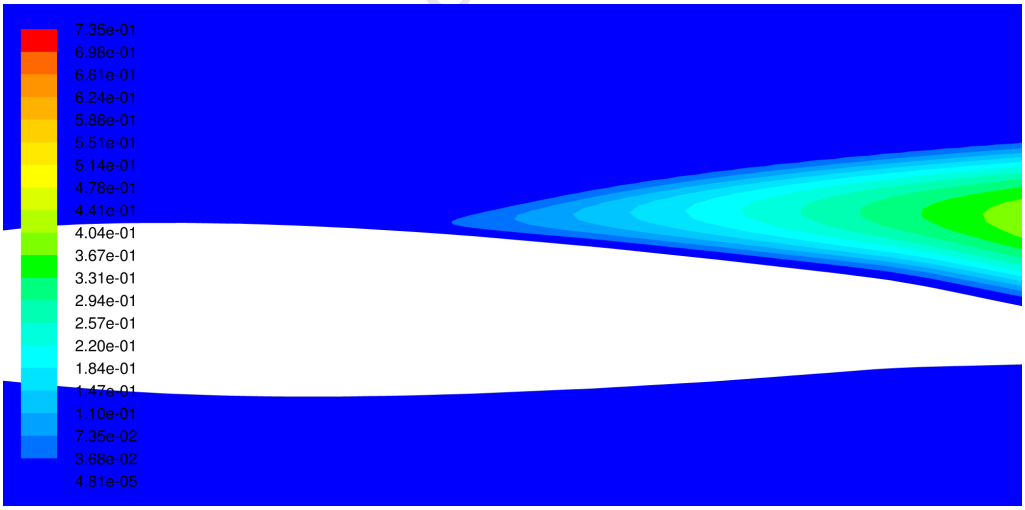


Figure 5.86: Close-up of region where turbulence is first experienced at $M_\infty = 0.8$, showing contours of effective viscosity

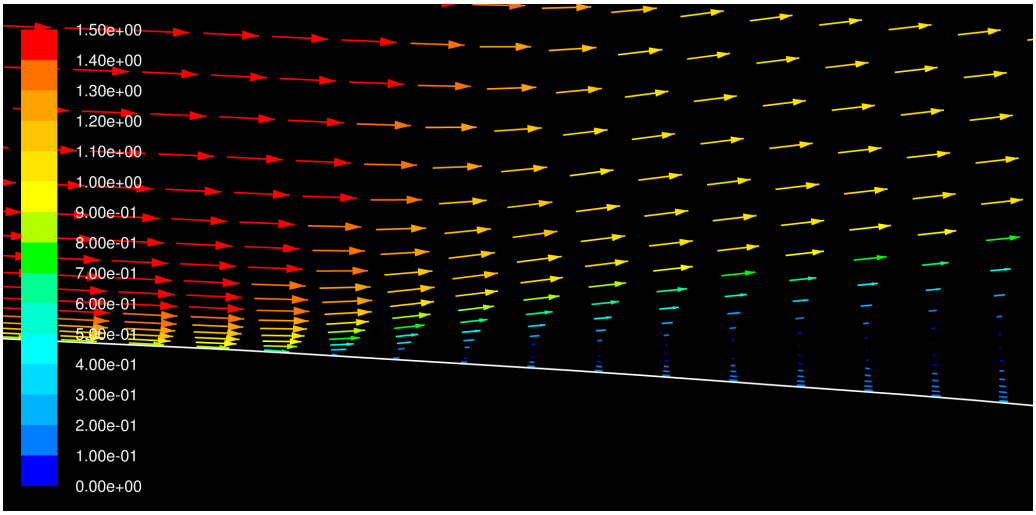


Figure 5.87: Close up of airfoil at the base of the shock for $M_\infty = 0.8$, showing vectors of Mach number

The delay of large-scale turbulent regions until the shock can also be seen for the case where $M_\infty = 0.7$ in Figs. 5.49 and 5.50 and slightly less obviously where $M_\infty = 0.6$ in Figs. 5.46 and 5.47. An examination of the area at the base of the shock in both cases shows separation where $M_\infty = 0.7$ (Fig. 5.88) but thickening of the boundary layer with no separation where $M_\infty = 0.6$ (Fig. 5.89).

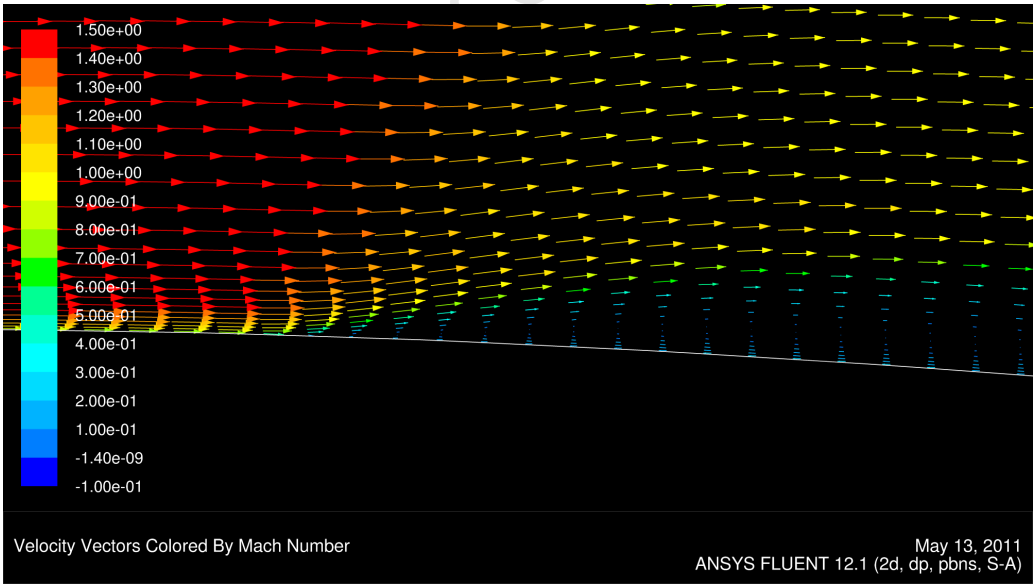


Figure 5.88: Close up of airfoil at the base of the shock for $M_\infty = 0.7$, showing vectors of Mach number

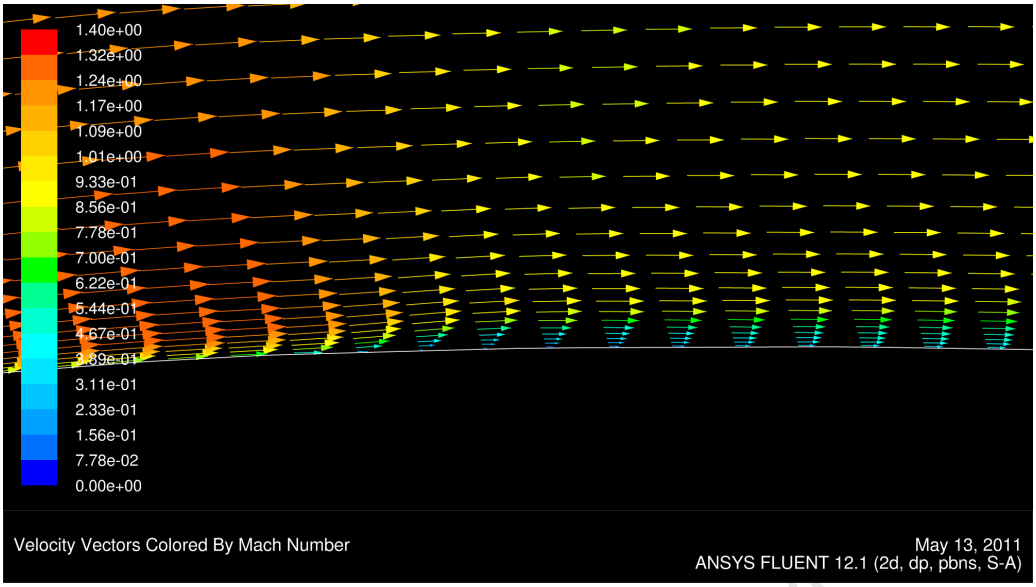


Figure 5.89: Close up of airfoil at the base of the shock for $M_\infty = 0.6$, showing vectors of Mach number

6 Modifications to PABLO

Pablo was modified to be used in the optimisation procedure, since its use of Matlab as a programming language made it able to be seamlessly integrated into the current framework of the project. The program is available as freeware and can be modified and used freely.

The aim here is to reduce the published version of PABLO to a single function which can be called once to yield all the necessary airfoil characteristics. These will be discussed in more detail in §10.4, but include the lift coefficient, C_L, as well as the hinge moment coefficients for the aileron and tab, C_{H_a} and C_{H_t}. Also, an additional feature to deflect the trailing edge had to be implemented so that a double-flap could be modeled. To allow for functionality, the "library file" option was retained from the original version as the means to input the specified profile, which refers to a text coordinate file. The vortex distribution was chosen as the preferred method for the panel-method solver. Here PABLO results using all three available methods were compared with XFoil (see Fig. 6.1), and although the vortex- and doublet-based distributions performed similarly, the vortex was chosen simply because it was used by XFoil.

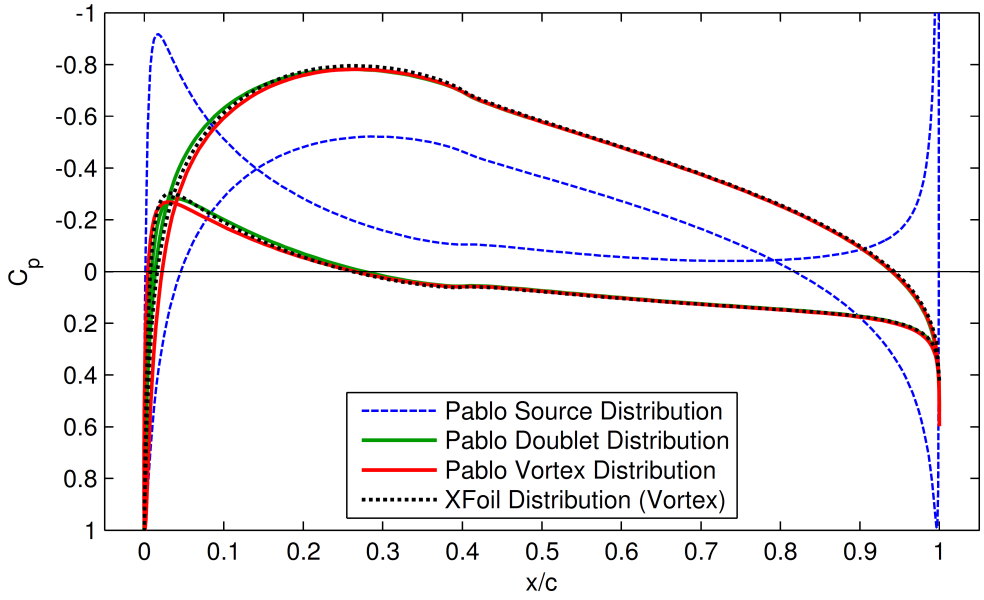


Figure 6.1: PABLO C_p distributions vs XFoil for an unflapped NACA 4412 airfoil at zero angle of attack

The following sections describe PABLOs main functions, detailing how they

implement a solution only where necessary or relevant. Each function's Matlab code appears in Appendix E. The modified functions can be directly compared with the original, which can be downloaded from `http://www.nada.kth.se/~chris/pablo/pablo.tar`. In the explanations that follow, bracketed numbers – for example "(15–25)" – refer to lines of code of the file in question as they appear in the appendix.

At the end of the chapter is a section where PABLO output variables C_L and C_H have been compared with outputs from XFoil's inviscid analysis, and the results show good correlation.

All input and output variables are listed and described in §6.9 on page 101.

6.1 `pablo.m`

The original main function file `pablo.m` was over 1000 lines long, as it included creation of the GUI, reading inputted information from the GUI, airfoil plotting procedures, and text output to the GUI, as well as the calling of functions to calculate airflow parameters. This was rewritten to exist as a small file, which simply calls functions in order to separate the various stages of the analysis. The Matlab `m`-file can be found in Appendix E.1, and performs the following tasks:

1. profile creation
2. calculation of surface pressure distribution
3. calculation of lift coefficient
4. calculation of hinge moments

Information for an end user is detailed below.

To call:

```
[c_l c_h_a c_h_t] = pablo(airfoil,x_a,y_a,x_t,y_t,alpha,delta_a,delta_t)
```

Inputs:

- airfoil profile `.data` filename (`airfoil`)
- hinge locations (`x_a`, `y_a`, `x_t`, `y_t`)
- angle of attack (`alpha`)
- aileron and tab flap deflection angles (`delta_a`, `delta_t`)

Outputs:

- lift coefficient (`c_l`)
- hinge moment coefficients (`c_h_a`, `c_h_t`)

6.2 `create_profile.m`

This is where the airfoil profile is created. The Matlab `m`-file can be found in Appendix E.2.

First the coordinate file is read in (3–14) and split into x and y vectors of each upper and lower surface (15–27). New x-vectors are created according to the amount of panels required, where the control of the node distribution in the x direction is passed to `distribute_x.m` (25–27), discussed further in §6.3. The upper and lower surfaces are then splined according to the new x-vector (33–42). Any trailing edge gap is closed (44–52), as the panel-method solver can only handle a sharp TE.

The profile coordinates are adjusted for the flap deflections by the function `rotate_hinge.m` (54–70), as will be discussed in §6.4. Due to the coding of this function the tab is deflected first, then the aileron, if both are deflections are indeed necessary. The airfoil is then scaled to ensure the chord is unity (72–79). Before program control is returned to `pablo.m` the node distribution is refreshed to ensure even distribution of the panels after the flap deflection procedure.

To call:

```
z = create_profile(nbpo2,airfoil,delta_a,delta_t,x_a,y_a,x_t,y_t)
```

Inputs:

- number of panels per surface (`nbpo2`)
- airfoil coordinate file (`airfoil`)
- flap deflections (`delta_a`, `delta_t`)
- hinge locations (`x_a`, `y_a`, `x_t`, `y_t`)

Outputs:

- `z`, an n-by-2 vector of airfoil coordinates, where $n-1$ is the total number of panels

6.3 `distribute_x.m`

This function creates a node distribution in the x direction according to the placement of the flap hinges. This ensures a greater node distribution around the sharp edges that occur in the vicinity of the hinge, allowing for greater accuracy in these regions. The Matlab `m`-file can be found in Appendix E.3.

There are four cases that may occur. In the case of neither flap being deflected, a standard *cosine spacing* is applied. For the two cases of either flap being deflected, the nodes are distributed so that a kind of modified cosine spacing is applied to each portion fore and aft of the hinge. Here the `flap_node_weight` variable (8 & 17) controls the amount of nodes in each section. For example, a

tab deflection only where $x_t = 0.95$ results in a 15%[1] node distribution aft of the hinge. Two separate half-cosine spaced x-vectors are created and then joined to create the complete vector. The final case is that of both flaps being deflected. Here the process is similar to that above, except three partitions are made, each with cosine spacing with a similarly weighted node distribution.

An example is shown in Fig. 6.2, where a profile with a single deflected control surface is compared with standard cosine spacing.

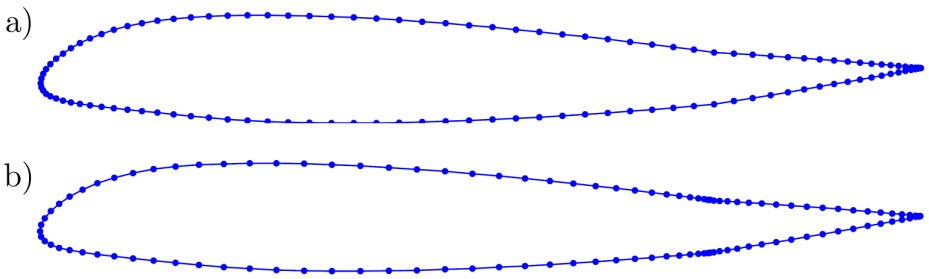


Figure 6.2: Node spacing for a 120-panel NACA 23012 airfoil with $\delta_a = 4$. Cosine spacing is applied a) to the profile as a whole, and b) both fore and aft of the hinge.

To call:

```
x = distribute_x(delta_a,delta_t,x_a,x_t,nbpo2)
```

Inputs:

- flap deflections (`delta_a`, `delta_t`)
- hinge $\frac{x}{c}$ locations (`x_a`, `x_t`)
- number of panels per surface (`nbpo2`)

Outputs:

- An $\frac{n}{2} + 1$ vector, where n is the total number of panels (`x`)

6.4 `rotate_hinge.m`

This function rotates the flap around the specified hinge by the specified deflection. The Matlab `m`-file can be found in Appendix E.4.

Figures 6.3 to 6.5 show the sequence of operations which are applied during this function. Figure 6.3 shows the absolute coordinates of the y hinge point located, calculated using the required relative position and interpolating where the upper and lower surface panels cut a vertical line through the hinge (14–25). A copy of the profile is made, and moved so the hinge coincides with the origin. It is then rotated about the origin through the specified deflection, and moved back so the hinge point corresponds to its original location (27–42). This results

[1] Here line 18 states `tail_nodes = (1-x_t)*flap_node_weight`, where `flap_node_weight` = 3 and `x_t` = 0.95, so `tail_nodes` = 0.15. In line 21 the nodes are distributed so that from the hinge to the training edge there are `nbpo2*tail_nodes` nodes.

in Fig. 6.4. The upper and lower intersection points are then found using the function `intersections`[2], and the profiles are joined appropriately, resulting in the deflected profile seen in Fig. 6.5. A new point is created at the point of intersection, as shown in Fig. 6.6, which prevents having panels of intermediate inclinations and sizes depending on the deflections. Although in some cases the new panels formed will be relatively tiny, the node refreshment detailed in the previous section will overwrite these new panels.

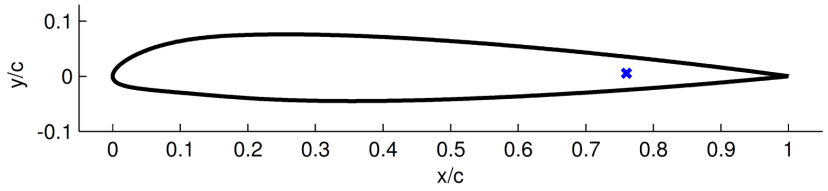


Figure 6.3: Flap deflection procedure: Step 1 – Locate the hinge in y

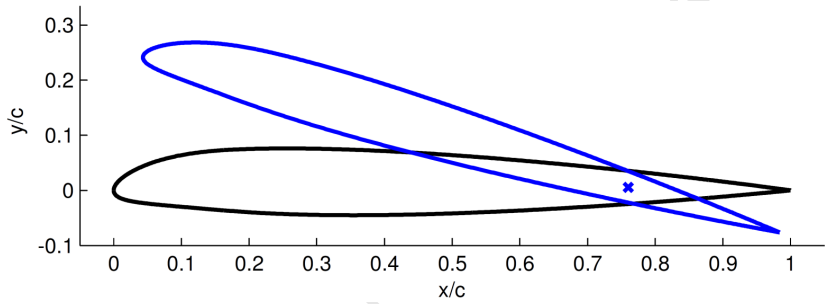


Figure 6.4: Flap deflection procedure: Step 2 – Rotate a copy of the profile

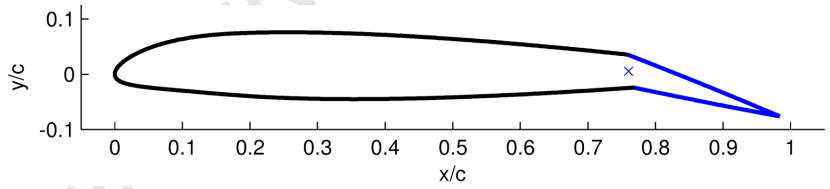


Figure 6.5: Flap deflection procedure: Step 3 - Trim the profiles

[2] This function was created by Douglas Schwarz and is available for dowloaded from `http://www.mathworks.com/matlabcentral/fileexchange/11837-fast-and-robust-curve-intersections`.

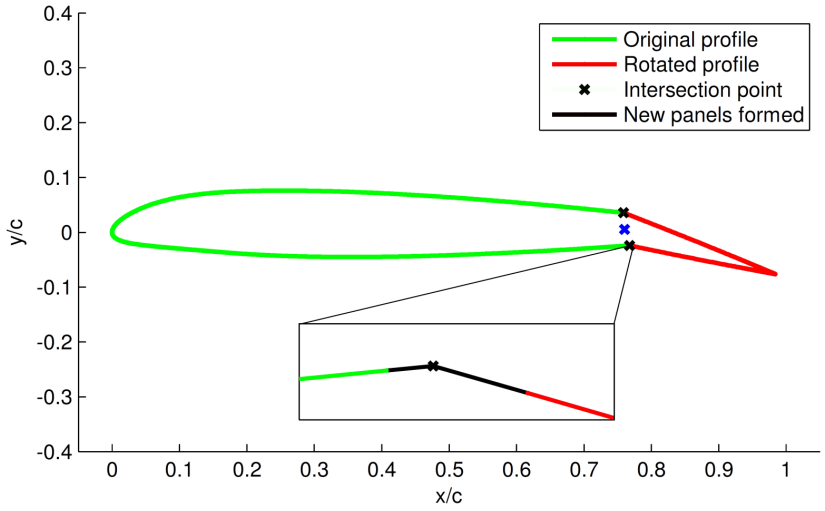


Figure 6.6: Flap deflection procedure: Detailing the addition of an intersection point during trimming procedure

To call:

```
function [xu yu xl yl] = rotate_hinge(delta,x_h,y_h,xu,xl,yu,yl)
```

Inputs:

- airfoil (x, y) coordinates (`xu`, `xl`, `yu`, `yl`)
- relevant hinge locations (`x_h`, `y_h`)
- flap deflection (`delta`)

Outputs:

- revised airfoil (x, y) coordinates (`xu`, `xl`, `yu`, `yl`)

6.5 `vortex.m`

The function `vortex.m` can be found in Appendix E.5 and contains the panel method code. It remains mostly unchanged, except for the outputs, where some characteristics were removed from the original and only the pressure distribution `cp` is returned.

The collocation points, where velocity is ultimately calculated, are situated midway between nodes. The pressure coefficient returned can therefore be seen as an average value over each panel.

To call:

```
function cp = vortex(za,alpha)
```

Inputs:

- airfoil (x, y) coordinates (`za`)
- angle of attack (`alpha`)

Outputs:

- pressure distribution (`cp`)

6.6 `lift.m`

The function `lift.m` can be found in Appendix E.6 and calculates the lift coefficient of the airfoil according to the equation 3.133 from [2]:

$$C_L = \frac{1}{c}\int_0^c C_{p_l}\mathrm{d}x - \frac{1}{c}\int_0^c C_{p_u}\mathrm{d}x$$

Because the function `atan2` (5–6) is used to calculate the four-quadrant panel angle this simplifies, with $c = 1$, to

$$C_L = -\oint C_p \mathrm{d}s$$

with s the curve traced out by the profile coordinates. With panel angle `theta` and angle of attack `alpha` this is coded (11–12) as

```
c_l = -trapz(Cp.*ds.*cos(theta-alpha))
```

with the built-in Matlab function `trapz` used as an approximation of the integral via the trapezoidal method.

To call:

```
function c_l = lift(Cp,z,alpha)
```

Inputs:

- distribution of pressure coefficient (`Cp`)
- airfoil (x, y) coordinates (`z`)
- angle of attack (`alpha`)

Outputs:

- lift coefficient per unit span (`c_l`)

6.7 `moment.m`

The function `moment.m` can be found in Appendix E.7. Here the hinge moment coefficient about the specified hinge is calculated in a similar way to the lift coefficient above, using the pressure distribution only from the x-position of the hinge point to the trailing edge, with the equation

$$C_H = \oint \left[\left(\frac{x}{c} - \frac{x_h}{c}\right) C_{p_x} + \left(\frac{y}{c} - \frac{y_h}{c}\right) C_{p_y}\right] \mathrm{d}\frac{s}{c}$$

with C_{p_x} and C_{p_y} the pressure coefficient in the x and y directions respectively. The line `Cp(x_mid<x_h) = 0;` ensures the above equation is zero along the curve for all $x < x_h$. **To call:**

```
function c_h = moment(Cp,z,x_h,y_h)
```

Inputs:

- distribution of pressure coefficient (`Cp`)
- airfoil (x, y) coordinates (`z`)
- relative hinge positions (`x_h`, `y_h`)

Outputs:

- hinge moment coefficient per unit span (`c_h`)

6.8 Verification with XFoil

After modification PABLO was checked against XFoil's inviscid solution. The results are shown in Figs. 6.7 and 6.8, showing a good match between the two solutions. Tests were performed on a NACA 23012 airfoil with $\delta_a = 4°$ and with the tab undeflected. The aileron hinge is at $x_h = 0.76$ and $y_h = 0.0053$ (or $0.5\frac{y}{c}$).

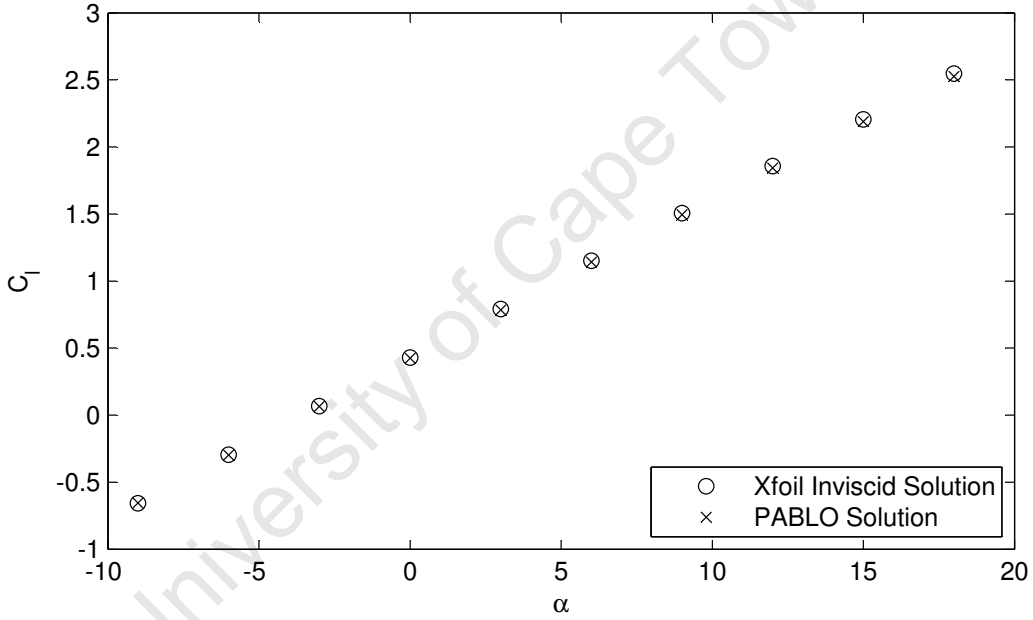


Figure 6.7: Lift coefficient of PABLO and XFoil for a range of α

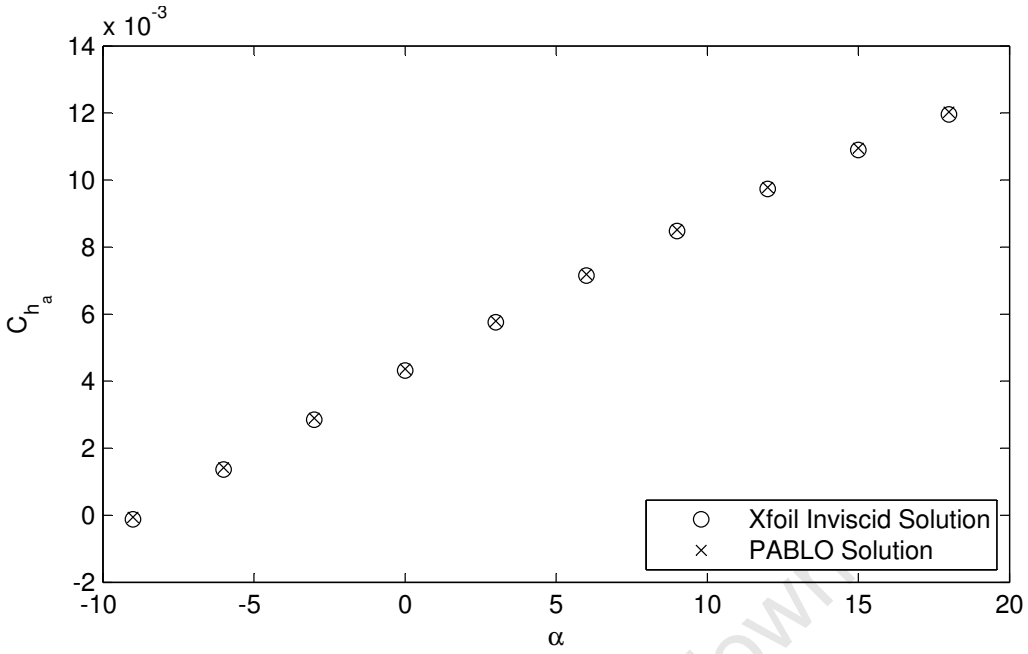


Figure 6.8: Hinge moment coefficient of PABLO and XFoil for a range of α

6.9 Pablo Variable Explanation

Here all the main input and output variables are explained, that are either needed to invoke PABLO, or which are passed between functions. An attempt has been made to present the variables in alphabetical order, but they are sometimes grouped for convenience. Local variables are not discussed here.

6.9.1 `airfoil`

This is the name of an ASCII file containing the airfoil profile. It requires a `.data` extension, and must form a continuous line from the trailing edge and back, going first over the top surface and back along the bottom. An example is shown below. There is no condition on number of points required.

```
  1:    # NACA 23012 airfoil. 280 nodes
  2:         1.000000    0.001260
  3:         0.996610    0.001810
  4:         0.991030    0.002710
  5:         0.984827    0.003704
  ⋮              ⋮           ⋮
142:         0.000217    0.002676
143:         0.000080    0.001609
144:         0.000010    0.000568
145:         0.000007   -0.000458
146:         0.000072   -0.001488
147:         0.000215   -0.002532
148:         0.000444   -0.003585
  ⋮              ⋮           ⋮
278:         0.984737   -0.003045
279:         0.990976   -0.002320
280:         0.996589   -0.001662
```

```
281:          1.000000   -0.001260
```

6.9.2 alpha

This is the angle of attack, to be input in degrees, and so that a positive angle of attack will cause an increase in lift.

6.9.3 c_h_a, c_h_t, c_h

These are the hinge moment coefficients for the aileron hinge (`c_h_a`), tab hinge (`c_h_t`) or a generic hinge (`c_h`).

6.9.4 c_l

This is the lift coefficient.

6.9.5 C_p

This is the pressure coefficient.

6.9.6 delta_a, delta_t, delta

These are the control surface deflections, to be input in degrees, and so that a positive deflection will cause an increase in lift. Variables used are for the aileron (`delta_a`), tab (`delta_t`), or a generic flap (`delta`) when a distinction need not be made.

6.9.7 nbpo2

A variable retained from the original program, this can be read as "number of panels over two," or the number of panels for each of the upper or lower surfaces. If the total number of panels for the computation is 280, `nbpo2` will equal 140.

6.9.8 x_a, y_a, x_t, y_t, x_h, y_h

These are the `x` and `y` hinge locations of the aileron (`_a`), tab(`_t`) or generic (`_h`) hinge. The x hinge locations are given as a ratio of $\frac{x}{c}$, with 0 at the LE and 1 at the TE. The y hinge locations are given as a ratio of the local $\frac{y}{t}$, with 0 at the lower surface and 1 at the upper. A graphical description is shown in Fig. 6.9 with an example in the caption.

6.9.9 x

This variable contains a vector of x-coordinates for the airfoil profile.

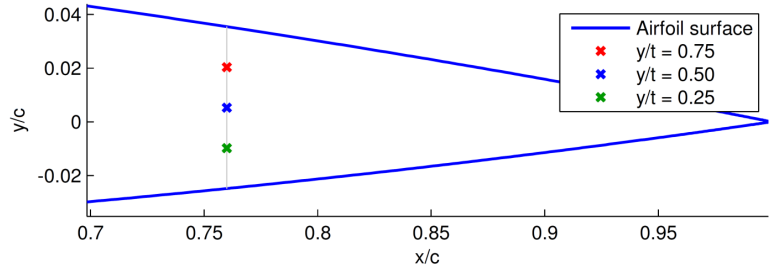


Figure 6.9: Hinge locations. The location of an aileron hinge at the red cross would be passed as `x_a = 0.76`, `y_a = 0.75`.

6.9.10 `xl`, `yl`, `xu`, `yu`

These variables contain the airfoil profile, and refer to the x and y vectors of the upper (`u`) and lower (`l`) surfaces, used when it is more convenient to pass the profile as separate vectors.

6.9.11 `z`, `za`

This variable in either form contains the airfoil profile as a single array, used when it is more convenient to pass the profile as a single unit.

Part II

Optimisation

7 Optimisation Introduction

The aim of the optimisation detailed in Part II of this dissertation is to evaluate the use of the modern design of experiments (MDOE) in determining the aileron-tab geometry which gives an optimum value for a particular merit function. Certain parameters will be varied between specified ranges, with a set of combinations performance-tested. Using a statistically-based approach, MDOE is then utilised to generate mathematical models to describe performance, which are then used to evaluate the merit function.

Although PABLO is used as the aerodynamic solver here, ultimately a more complex analysis will need to be done using CFD, where a single configuration may take up to an hour to analyse. MDOE is therefore used for its efficiency, requiring analysis of only a small number of configurations to be tested. To ensure there is no great loss of accuracy, results from the reduced MDOE data set will be compared with a full data set.

Although PABLO was seen to provide results that are insufficient for a realistic analysis of transonic flow conditions, the comparison between optimisation methods here should be unaffected by this restriction. Here PABLO was chosen for its speed and its seamless integration into an optimisation procedure that was to be written in Matlab.

The methods of design of experiments (DOE) have been successfully employed for this kind of task in aerodynamics as well as other fields where a product or process is systematically improved or optimised. This chapter will cover these methods in a broad sense and detail certain aspects which will be used during the optimisation procedure.

This part begins with a review of the theory behind the MDOE method. The methodology used to garner MDOE results are then outlined, with a few preliminary results given. A full analysis of the merit function follows. Finally some concluding remarks are made.

8 Optimisation Theory

8.1 Basic Principles of Modern Design of Experiments

Summarised from [4] and [66]

The methods of Modern Design of Experiments (MDOE) use statistical analyses to arrange and plan experimental programmes in order to achieve two primary objectives:

1. to experimentally establish a mathematical description of the desired phenomena
2. to most efficiently find a design optimum

MDOE can be applied to any process having clearly defined inputs that can be varied to cause fluctuation of a measurable output quantity. Figure 8.1 shows the traditional Design of Experiments flow of inputs and outputs. These inputs are called factors and the output is the response. Factors can be numeric or categoric, with categoric factors describing alternatives like, for example, "Catalyst A" and "Catalyst B". The inputs and outputs in Fig. 8.1 are described below [66].

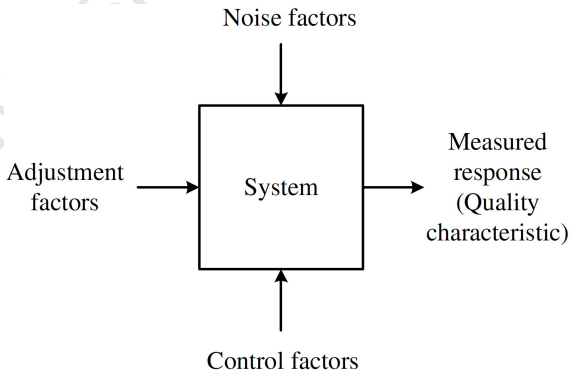


Figure 8.1: The Parameter Diagram, or P-Diagram, introduced by Madhav Phadke

Adjustment Factors

The parameters set by the end user or operator to vary the desired response of the product.

Control Factors

The design characteristics that are free to be adjusted during the design process. These are the parameters that are adjusted to achieve an optimised product or process.

Noise Factors

Anything that cannot be controlled by the designer that causes the output response to deviate from its target value.

Quality Characteristic

The measured output response of a design.

MDOE uses arrays to plan experimental programmes set in a statistical framework to gather information from the results in a methodical way. This follows the founding work of Sir Ronald Fischer in the 1920s and many advancements made since [66]. The methods used will be discussed in the following sections.

8.2 Types of Experimental Arrays

Summarised from [66]

In traditional experimental design certain levels are chosen for each factor, and a number of experiments are performed where the factors are varied according to a specific programme while the response is recorded. The combination of factor levels set during testing are represented by each row in a test matrix; a row is sometimes called a run or a treatment combination. Several different types of test matrices exist, some of which will be explored here.

Full factorial

A full factorial test matrix explores every possible combination of factor levels, as shown in Table 8.1 for a four-factor experiment where each factor, F, has two levels, with –1 and +1 indicating the lower and higher levels respectively. The full factorial programme is the most comprehensive and therefore most costly, although the trade off is that it can yield more accurate information than using matrices of reduced size.

Table 8.1: Experimental runs for a full-factorial test programme

Run	F_1	F_2	F_3	F_4
1	−1	−1	−1	−1
2	−1	−1	−1	+1
3	−1	−1	+1	−1
4	−1	−1	+1	+1
5	−1	+1	−1	−1
6	−1	+1	−1	+1
7	−1	+1	+1	−1

Continued on next page...

Table 8.1: ... continued

Run	F_1	F_2	F_3	F_4
8	−1	+1	+1	+1
9	+1	−1	−1	−1
10	+1	−1	−1	+1
11	+1	−1	+1	−1
12	+1	−1	+1	+1
13	+1	+1	−1	−1
14	+1	+1	−1	+1
15	+1	+1	+1	−1
16	+1	+1	+1	+1

One factor at a time

The test matrix below uses a one factor at a time (OFAT) procedure for the same factors and levels as before. Although the one factor at a time approach is sometimes efficient, and gives some idea of the variation a control factor can produce, it does not provide information on interactions between factors.

Interactions are important phenomena that can cause the variation of one factor to have a different effect depending on the level of one or more other factors. They are described in more detail in §8.4.

Table 8.2: Test matrix for an experimental programme.

Run	F_1	F_2	F_3	F_4
1	−1	−1	−1	−1
2	−1	−1	−1	+1
3	−1	−1	+1	+1
4	−1	+1	+1	+1
5	+1	+1	+1	+1

Fractional factorial/Orthogonal array

The family of fractional factorial arrays use factor combinations such that each factor, as well as each combination of factor, is represented an equal number of times, as is also the case for the full-factorial array. This gives them the property of being orthogonal, which can be most easily seen for two-level factor arrays where the dot product of any two columns is zero, as for that shown in Table 8.3.

Table 8.3: Experimental runs for a fractional-factorial programme

Run	F_1	F_2	F_3	F_4
1	−1	−1	−1	−1
2	−1	−1	−1	+1
3	−1	+1	+1	−1
4	−1	+1	+1	+1
5	+1	−1	+1	+1
6	+1	−1	+1	−1
7	+1	+1	−1	+1
8	+1	+1	−1	−1

To compare the efficiency of fractional-factorial over full-factorial arrays, the 50% reduction in runs we see here (compared with Table 8.1) is very modest. Other configurations allow for better results. For example one orthogonal array, the so-called $L_{36}\left(2^{11} \times 3^{12}\right)$, has 36 test runs accommodating up to 11 2-level factors and 12 3-level factors. This would require $2^{11} \times 3^{12} = 1,088,391,168$ runs if a full factorial test matrix were to be used.

MDOE employs the use of these fractional-factorial arrays, which can reputedly give results of similar accuracy to full-factorial arrays, but at a small fraction of the experimental cost. This is because the measured data points are more valuable; since more than one factor is changed at a time between each run more information is contained in the results. The data can later be analysed and the effects of each factor partitioned using the analysis of variance (ANOVA). It is the orthogonality of the test matrix which allows for this partitioning of effects, which will be discussed in the following section.

8.3 Factor Effects

This section summarised from §3.3 of [66]

The relative effect each factor has on the response can be compared qualitatively by using an additive model, represented by

$$\eta(F_{1_h},\ F_{2_i},\ F_{3_j},\ F_{4_k}) = \mu + a_h + b_i + c_j + d_k + e$$

Here μ is the mean value of the response η, e is the residual, and the terms a_h, b_i, c_j and d_k are the deviations from the mean caused by setting the factors F_1, F_2, F_3 and F_4 at levels h, i, j and k respectfully.

We can calculate the effect on the response caused by varying a certain factor between the prescribed levels by averaging all values of the response where the

factor was at that level. Using the test matrix of Table 8.3 as an example, we can calculate $m_{F_{3_{-1}}}$, which is given by

$$\begin{aligned} m_{F_{3_{-1}}} &= \frac{1}{4}(\eta_1 + \eta_2 + \eta_7 + \eta_8) \\ &= \frac{1}{4}[(\mu + a_{-1} + b_{-1} + c_{-1} + d_{-1}) + e_1 \\ &\quad + (\mu + a_{-1} + b_{-1} + c_{-1} + d_{+1}) + e_2 \\ &\quad + (\mu + a_{+1} + b_{+1} + c_{-1} + d_{+1}) + e_7 \\ &\quad + (\mu + a_{+1} + b_{+1} + c_{-1} + d_{-1}) + e_8] \\ &= \frac{1}{4}(4\mu + 4c_{-1}) + \frac{1}{4}(2a_{-1} + 2a_{+1}) + \frac{1}{4}(2b_{-1} + 2b_{+1}) \\ &\quad + \frac{1}{4}(2d_{-1} + 2d_{+1}) + e_1 + e_2 + e_7 + e_8 \end{aligned}$$

And since, by definition,

$$a_{-1} + a_{+1} = b_{-1} + b_{+1} = d_{-1} + d_{+1} = 0$$

as a_{-1} and a_{+1} are deviations from the mean caused by factor a (and similarly for b and d), this yields

$$m_{F_{3,\,-1}} = (\mu + c_{-1}) + \frac{1}{4}(e_1 + e_2 + e_7 + e_8)$$

Therefore $m_{F_{3_{-1}}}$ is an estimate of $\mu + c_{-1}$.

The mean values of the response for each factor level can be calculated in this way to construct a graph like that shown in Fig. 8.2, as a purely hypothetical example. Here each factor is plotted in turn. They all have the same mean, and it can be easily seen which level provides the optimum for maximisation/minimisation of the response, as well as the relative magnitude of the effects of each factor. Here, to maximise η, F_1, F_2, F_3 and F_4 would need to be set at levels 1, -1, -1 and 1 respectfully. Since this configuration does not appear in Table 8.3 the initial experiment would not reveal this optimum. F_3 clearly has the greatest effect on the response, while F_4 has the smallest effect.

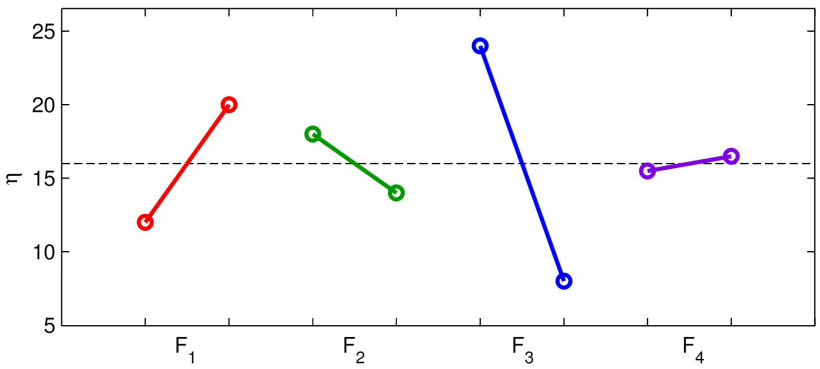


Figure 8.2: Example of how factor effects may be qualitatively analysed

8.4 Interactions

This section summarised from [66]

Interactions can also be studied using the additive model from the previous section. Interactions were described earlier as when changing one factor's level, the effect on the response might be different according to the setting of another factor. These can also be graphically analysed, examples of which are shown in Fig. 8.3. Here two 3-level factors, A and B, are shown. In a) there is no interaction, as evidenced by the parallel lines, where the setting of B changes the response, but does not change the manner in which A affects the response. In b) there is a synergistic interaction, where, as B changes, the increase in the response by changing A is amplified. And c) shows an antisynergistic interaction where changing B causes A to have a dramatically different effect, effectively changing the fundamental way that A affects the response. Although the interactions shown here are between two factors, they can also occur between three or more factors.

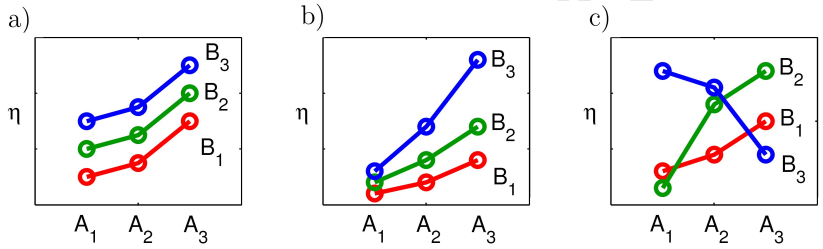


Figure 8.3: Analysis of interactions a) none, b) synergistic, and c) antisynergistic

For the plots in Fig. 8.3, the average values of the response are calculated for each combination of factors under investigation. So, for example, the value of η on the ordinate of the interaction plots corresponding to A1-B1 are simply the mean of all the responses that had A and B at the lower levels. Interactions can be investigated in this way, but for a more objective approach statistical methods are used, as will be described in §8.9.

8.5 Calculating Degrees of Freedom

This section summarised from [66]

The minimum number of runs required for any matrix experiment can be calculated from the total degrees of freedom. In general the degrees of freedom associated with a factor is equal to one less than the number of levels for that factor, while the degrees of freedom of an interaction is equal to the product of the degrees of freedom for each of the factors involved in the interaction. One degree of freedom is always associated with the mean.

The minimum number of runs for a test having 7 3-level factors where 9 interactions need to be examined is given by Table 8.4.

Table 8.4: Calculating minimum runs from degrees of freedom.

Source	Degrees of freedom
Mean	1
Factors	$7 \times (3-1) = 14$
Interactions	$9 \times (3-1) \times (3-1) = 36$
Total	51

The number of degrees of freedom available in a particular test array is given by the degrees of freedom available in each column. Each column has the same degrees of freedom than the factor it can accommodate, or one less than the number of levels of the factor.

As an example, an L_{27} (3^{13}) test matrix can accommodate 13 three-level factors, with each column having two degrees of freedom. The total degrees of freedom available is therefore 26. If interactions are to be analysed two columns will be needed to accommodate each interaction.

8.6 Confounding: Linear Graphs and Interaction Tables

Summarised from [66] and [67]

Fractional factorial arrays have a property where certain interactions will be *confounded* with the main factor effects. When setting up a test matrix and assigning factors to columns of an orthogonal array, great care must be taken to ensure that those factors which have significant interactions are in appropriate columns to avoid this aliasing of effects.

To illustrate why the confounding of factors and interactions occurs the process of construction of these array must be examined. The base of the L_8 array shown in Table 8.5 is the 2^3 full factorial array of Table 8.6. Using brackets to represent column numbers, it can be seen that (1), (2), and (3) of the full array fit into (1), (2), and (4) of the Taguchi array. Now to generate the remaining columns the existing columns of the 2^3 full array are used. Working only in the Taguchi array, (3) is simply $(1)+(2)$ modulus 2, (5) is $(1)+(4)$ modulus 2, (6) is $(2)+(4)$ modulus 2, and (7) is $(1)+(2)+(4)$ modulus 2.

Due to this linear dependence between columns the effect of any associated interaction will not be able to be separated from the effect of a factor placed in the corresponding column, and as such confounding will occur. Of course if the interaction is insignificant the column will be available for use, although this should be avoided if possible.

Table 8.5: Taguchi $L_8(2^7)$ array

(1)	(2)	(3)	(4)	(5)	(6)	(7)
0	0	0	0	0	0	0
0	0	0	1	1	1	1
0	1	1	0	0	1	1
0	1	1	1	1	0	0
1	0	1	0	1	0	1
1	0	1	1	0	1	0
1	1	0	0	1	1	0
1	1	0	1	0	0	1

Table 8.6: 2^3 full factorial array

(1)	(2)	(3)
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

As an example of how confounding affects the ability to distinguish between the cause of a change of the response, consider factors A and B placed in (1) and (2) of Table 8.5 with factor C in (3). C_1 occurs when A and B are simultaneously at their lower and higher levels A_1B_1 and A_2B_2, and C_2 occurs at the mixed levels A_1B_2 and A_2B_1. An AB interaction is shown in Fig. 8.4, where the magnitude of the interaction is measured by the extent of non-parallelism, given by

$$\begin{aligned} AB &= (y_{A_2B_2} - y_{A_1B_2}) - (y_{A_2B_1} - y_{A_1B_1}) \\ &= (y_{A_2B_2} + y_{A_1B_1}) - (y_{A_2B_1} + y_{A_1B_1}) \\ &= 2C_1 - 2C_2 \end{aligned} \tag{8.1}$$

This shows that the interaction AB is measured by the change in factor C, so the effect of the AB interaction and of the factor C cannot be separated.

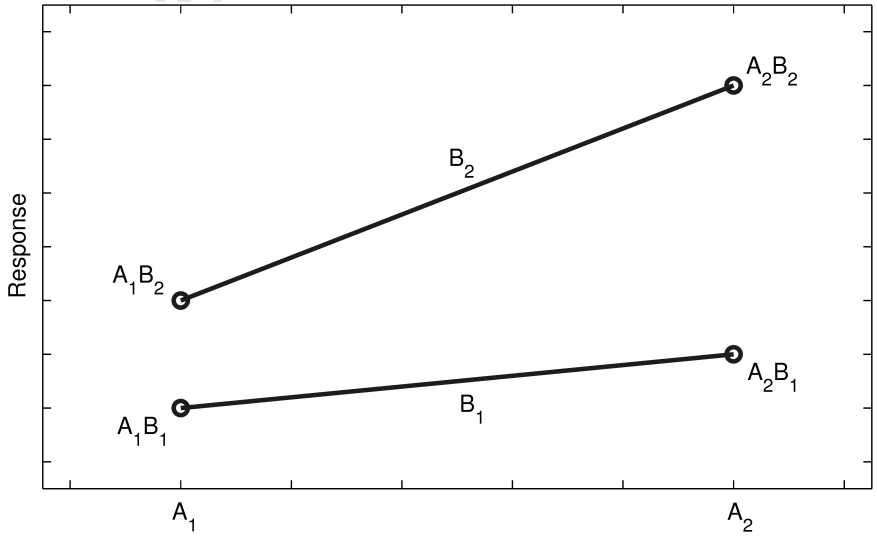


Figure 8.4: Interaction between A and B leading to possible confounding.

Due to the previous discussion, an interaction table exists for each orthogonal array, as shown in Table 8.7 for the $L_8(2^7)$ array. Any two columns containing

factors involved in an interaction can be read to see which column must be left empty to avoid confounding. This can place a great restriction on these arrays if multiple interactions are to be investigated.

Table 8.7: Interaction table for L_8 (2^7) orthogonal array

	(1)	**(2)**	**(3)**	**(4)**	**(5)**	**(6)**	**(7)**
(1)	–	3	2	5	4	7	6
(2)		–	1	6	7	4	5
(3)			–	7	6	5	4
(4)				–	1	2	3
(5)					–	3	2
(6)						–	1
(7)							–

Linear graphs can also be consulted for correct placement of factors into columns in an array. The two linear graphs that exist for the L_8 (2^7) array are shown in Fig. 8.5. The columns of the array are represented by the lines and dots, where the column represented by the line is confounded with the columns which label the dots at either end. In this case both (or generally all) linear graphs for a particular array must be consulted to avoid confounding.

Figure 8.5: Linear graphs for L_8 (2^7) orthogonal array

8.7 Building an Empirical Model

This section summarised from [4]

Having gathered data of a measured response from an experiment using an orthogonal array, the response can be approximated by a linear function. A first order model of the example shown in Table 8.3 would be described by

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \varepsilon$$

where β represents the coefficients of the respective "regressors", x_i, which in turn describe the levels of factor F_i but are converted to "coded" variables to range

between -1 and 1. Variables are coded according to the equation

$$x_{ij} = \frac{F_{i,\,j} - \left[\max\left(F_i\right) + \min\left(F_i\right)\right]/2}{\left[\max\left(F_i\right) - \min\left(F_i\right)\right]/2}$$

for $1 \leq i \leq k$ and $1 \leq j \leq \ell$, for a model with k regressors having ℓ levels.

The use of coded variables decouples the y-intercept. The coding of variables also prevents any rounding errors, which may be prevalent when using factors having largely different magnitudes. ε represents the error in the approximation, which can be thought of as the higher order terms of what is essentially a first-order Taylor series expansion in the example above. Here the β_is are solved for to determine the mathematical model, which would need to be converted from the coded variables back to the natural variables in order to describe the actual values of each factor.

A more complex model that includes interactions and higher order terms can be included while maintaining the linearity of the actual regression. So a term $\beta_5 x_5$ may be introduced, where x_5 could, for example, represent $x_1 x_3$ or x_2^2. The model would be second-order in x but still linear in β.

An array of regressors, $\boldsymbol{X}$, is set up to form the matrix equation

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

where

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_k \end{bmatrix}, \quad \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

The method of least squares is used to obtain an estimator of $\boldsymbol{\beta}$. Here the sum of squares of the deviation of the regression model to the response is minimised. That is, we wish to find values of $\boldsymbol{\beta}$ that minimise

$$L = \sum_{i=1}^{n} \boldsymbol{\varepsilon}_i^2 \tag{8.2}$$

The least squares estimator of $\boldsymbol{\beta}$ is given by $\boldsymbol{b}$ and found by setting $\frac{\partial L}{\partial \boldsymbol{\beta}} = 0$, is given by

$$\boldsymbol{b} = \left(\boldsymbol{X}'\boldsymbol{X}\right)^{-1}\boldsymbol{X}'\boldsymbol{y} \tag{8.3}$$

The fitted regression model is given by

$$\hat{\boldsymbol{y}} = \boldsymbol{X}\boldsymbol{b} \tag{8.4}$$

where the difference between the actual (recorded) response and the predicted

response is the residual, given by

$$\boldsymbol{e} = \boldsymbol{y} - \hat{\boldsymbol{y}} \tag{8.5}$$

Here $\boldsymbol{X}$ is called the design matrix, since it is generally unique to the particular mathematical model chosen. The values of each element are dictated by the levels of the factors contained in the corresponding term for that particular treatment combination. For example, the interaction term x_1x_2 would be represented by 1 in the design matrix if x_1 and x_3 were both either 1 or -1. If this interaction were contained in the term β_5x_5 as in the discussion earlier, then the value would essentially be the coefficient of the unknown β_5 and as such appear in the sixth column of the design matrix (being that the intercept, given by β_0, takes up the first column).

8.8 Analysis of Variance

This section summarised from [66] and [68]

Analysis of variance (ANOVA) can be used to give a measure of the importance of each regressor so that an adequate mathematical model can be found. It essentially tells us how much of the variation in the measured response is explained by each included regressor. Random-effects ANOVA is used for models that contain quantitative factors, where the levels chosen during testing are a sample of a general population. This is distinct from fixed-effects ANOVA where each factor can either be qualitative, or if it is quantitative then only those levels being tested can be considered for the solution. To help refine a model hypothesis testing is used to determine the value of a regressor or group of regressors. Here the extra sum of squares principle is used.

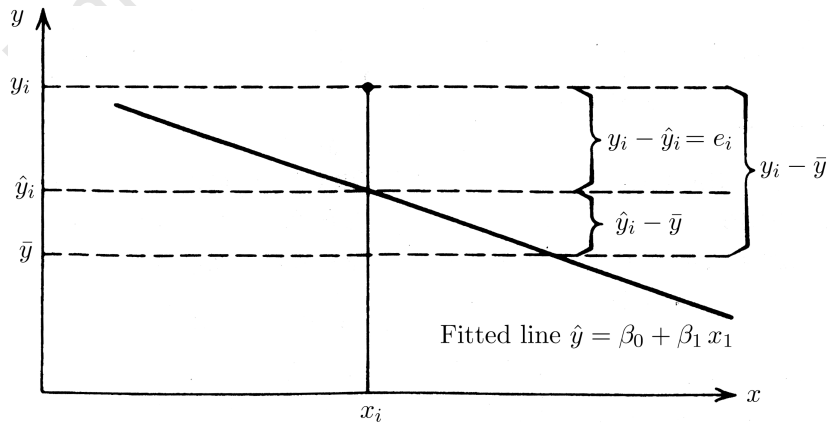


Figure 8.6: Geometric interpretation of the sum of squares identity, equation 8.6 [68]

To understand the sum of squares, we need to examine the differences between the mean, the measured response, and the predicted response, shown in Fig. 8.6

as $\bar{y}$, y_i, and $\hat{y}_i$ respectively for one given x_i. This gives the identity

$$(y_i - \bar{y}) = (\hat{y}_i - \bar{y}) + (y_i - \hat{y}_i) \tag{8.6}$$

Squaring each term and and summing over all values of x yields an identity for the sum of squares

$$\sum (y_i - \bar{y})^2 = \sum (\hat{y}_i - \bar{y})^2 + \sum (y_i - \hat{y})^2 \tag{8.7}$$

The left hand side of the above equation is the sum of squares about the mean, or total sum of squares, SS_t, which is the total variation in the response from the mean. The total sum of squares can then be partitioned into an "explained" component and an "unexplained" component. The first term on the RHS of equation 8.7 is the sum of squares due to regression, or the regression sum of squares, SS_r. Being the variation between the predicted response and the mean, it is considered to be "explained" by the chosen model. The remaining variation that remains "unexplained" is the error sum of squares, SS_e, and is represented by the residual. This is the difference between the measured response and the predicted values of the response.

The mean sum of squares are computed by dividing the sum of squares by the degree of freedom for that sum. For a model with no regressors the sum of squares has $n - 1$ degrees of freedom, since having the sum and $n - 1$ values of the response, the nth response can be calculated. Here the total sum of squares and the error sum of squares are equivalent. As regressors are added, each having 1 degree of freedom, the error sum of squares degrees of freedom decreases by the amount of regressors, k. Formulae for the mean sum of squares are then

$$MS_e = \frac{SS_e}{n - k - 1}$$

and

$$MS_r = \frac{SS_r}{k}$$

ANOVA results are usually tabulated, and will be further explained in §8.10.

8.9 Hypothesis Testing

This section summarised from [4]

The sum of squares can be used to test for the significance of single regressors, although this method is equally valid for testing groups of regressors. The addition of regressors, in all cases, will explain more of the total variance and therefore reduce SS_e and increase SS_r by a certain amount. This amount is recorded by the F-statistic, which is evaluated using a z-distribution table to determine if the additional regressors are statistically significant.

Consider the regression model

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \tag{8.8}$$

To determine if a subset of the k regressors contribute significantly to the model we partition the vector of regression coefficients so that

$$\boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta_1} \\ \boldsymbol{\beta_2} \end{bmatrix}$$

where $\boldsymbol{\beta}_1$ contains the regressor or subset of regressors to be examined. The model can then be written

$$\boldsymbol{y} = \boldsymbol{X_1}\boldsymbol{\beta_1} + \boldsymbol{X_2}\boldsymbol{\beta_2} + \boldsymbol{\varepsilon}$$

and we test the hypotheses

$$\begin{aligned} H_0 &: \beta_1 = 0 \\ H_1 &: \beta_1 \neq 0 \end{aligned} \tag{8.9}$$

If the null hypothesis H_0 is not rejected then the regressors associated with the β_1 coefficients are all zero and the regressors can be deleted from the model.

The F-statistic used to evaluate H_0 is given by

$$F_0 = \frac{\left[SS_r(\boldsymbol{\beta}) - SS_r(\boldsymbol{\beta}_2)\right]/r}{MS_e}$$

where r is the amount of regressors in $\boldsymbol{\beta}_1$ and MS_e is calculated for the full model. This essentially calculates the change in the explained sum of squares due to inclusion of the specified regressors. If F_0 is appreciably small H_0 is not rejected. The limiting value of F_0 is found in tables and given by $F_{\alpha,\, r,\, n-p}$, where α is a chosen significance level and $p = k+1$. If the computed F-statistic is greater than $F_{\alpha,\, r,\, n-p}$ for an α of 0.05, then we can say with at least a 95% confidence that the terms in $\boldsymbol{\beta}_1$ are significant.

In addition to the F-test, we can perform a t-test to test the null hypothesis on individual regressors. The principle is the same as the F-statistic, but here we apply the hypothesis to single regressors, where

$$\begin{aligned} H_0 &: \beta_j = 0 \\ H_1 &: \beta_j \neq 0 \end{aligned} \tag{8.10}$$

The test statistic is given by

$$t_0 = \frac{b_j}{\sqrt{\hat{\sigma}^2 C_{jj}}} \tag{8.11}$$

Here $\hat{\sigma}^2$ is the standard deviation and C_{jj} is the diagonal element of $(\boldsymbol{X}'\boldsymbol{X})^{-1}$ corresponding to β_j. H_0 is rejected if $t_0 > t_{\alpha/2,\, n-k-1}$, where again this limiting value of t is found from tables and represents the $100\,(1-\alpha)\%$ probability that such a value of t_0 would be due to chance alone.

The F- and t-statistics are related by the identity $F = t^2$ when the F-test is applied to a single variable.

8.10 Tabulating ANOVA Results

This section summarised from [4]

A typical ANOVA table is shown below, where the p-values are also computed from tables using the F-statistic. The p-value tells us the probability of having the calculated F statistic due to random fluctuations, where a value of p of, say, 0.01, would indicate there is only a 1% chance of seeing such a high F simply due to chance and therefore a great probability that the evaluated regressors are significant. Smaller values of p therefore indicate significant model terms.

Table 8.8: Typical layout of ANOVA table [4]

Source of Variation	**Sum of Squares**	**Degrees of Freedom**	**Mean Square**	$\boldsymbol{F_0}$	$\boldsymbol{p}$**-value**
Model	SS_r	k	MS_R	MS_R/MS_e	*from table*
Error on Residual	SS_e	$n-k-1$	MS_e		
Total	SS_t	$n-1$			

8.11 Standard Mathematical Models

This section summarised from [4]

Listed here are typical standard mathematical models used to choose regressors.

Mean

This very basic model is comprised of the intercept only.

$$y_{\text{mean}} = \beta_0$$

Linear

This contains terms that are linear in x as well as the intercept.

$$y_{\text{linear}} = y_{\text{mean}} + \beta_i\, x_i \quad \text{for } 1 < i < k$$

2FI

This model contains the same terms as the linear model, with the addition of two-factor interactions.

$$y_{\text{2FI}} = y_{\text{linear}} + \beta_i\, x_m x_n$$

$$\text{for } k+1 < i < \frac{k!}{k} + k + 1, \quad 1 < m < k, \quad 1 < n < k, \quad m < n$$

Quadratic

This model contains the same terms as the 2FI model, with the addition of squared terms.

$$y_{\text{quadratic}} = y_{\text{2FI}} + \beta_i \, x_j^2 \quad \text{for } \frac{k!}{k} + k + 2 < i < \frac{k!}{k} + 2k + 2 \;\; 1 < j < k$$

Some less structured models exist, and will be discussed in the following section.

8.12 Model Refining Methods

This section summarised from [4]

A regression model must be chosen that best represents the measured response. No unique procedure exists for selecting the best regression terms but instead there are several, although these may not ultimately select the same model. These will be briefly discussed here.

All possible regressions

As the name suggests every possible combination of regressors are examined and evaluated. This can be very time consuming for experiments with more than a few regressors and especially if the inclusion of higher order terms is to be considered. The number of possible combinations is given by

$$\sum_{k=0}^{n} \frac{n!}{(n-k)!k!}$$

so a quadratic model having 5 factors with a total of 25 regressors would yield 33,554,432 possible model combinations.

Forward substitution

A model is built up by adding candidate regressors to a simple base model. Hypothesis tests are carried out on each candidate regressor, adding the most significant one to the model in turn until no more significant regressors exist.

Backwards elimination

Here the process starts with a comprehensive model and existing regressors are evaluated, with the least significant being eliminated one at a time as constant reevaluation of the updated model takes place.

Stepwise regression

This is a mix of the previous two methods. From a basic model candidate regressors are added if they are found to be significant. After each addition the

model is reassessed to check if the recent addition has rendered any existing regressors nonsignificant, as is sometimes the case.

8.13 Scaled Residuals

This section summarised from [4]

There are several ways to scale residuals so that they convey more information than the ordinary least squares residuals of equation 8.5. Several will be described here.

8.13.1 Standardised Residuals

The standardised residual is given by

$$d_i = \frac{e_i}{\hat{\sigma}}, \quad i = 1,\, 2, \ldots, n$$

where $\hat{\sigma}$ can be estimated by $\sqrt{MS_e}$, which is generally used for computation. These residuals have a zero mean like ordinary residuals, but they have approximately unit variance. Standardised residuals are therefore useful for identifying outliers, which are considered to be all those that have values outside $-3 \geq d_i \geq 3$. Outliers may represent a region in the variable space where the model does not approximate the true response very well.

8.13.2 Studentised Residuals

Studentised residuals are given by

$$r_i = \frac{e_i}{\sqrt{\hat{\sigma}^2\,(1 - h_{ii})}}, \quad i = 1,\, 2, \ldots, n$$

Here $\hat{\sigma}^2 = MS_e$, and h_{ii} are the diagonal elements of $\boldsymbol{H}$, the so-called *hat* matrix, since it transforms the measured response y to the fitted response $\hat{y}$ as seen in equation 8.12.

$$\begin{aligned} \hat{\boldsymbol{y}} &= \boldsymbol{X}\boldsymbol{b} \\ &= \boldsymbol{X}\left(\boldsymbol{X}'\boldsymbol{X}\right)^{-1}\boldsymbol{X}'\boldsymbol{y} \\ &= \boldsymbol{H}\boldsymbol{y} \end{aligned} \tag{8.12}$$

In some data sets the standard deviation may differ greatly. The studentised residuals take this into account, whereas the standardised residuals use the average of the standard deviation. Although standardised and studentised residuals are often quite similar, especially for large data sets, it is generally recommended to use studentised residuals.

8.13.3 PRESS Residuals and the PRESS Statistic

To calculate the prediction sum of squares (PRESS) residual of the ith observation the regression model is fitted to the remaining $n-1$ points yielding the ith point. The predicted value is denoted $\hat{y}_{(i)}$ and the prediction error is $e_{(i)} = y_i - \hat{y}_{(i)}$. Fortunately one does not need to do this full computation for each data point, as the PRESS residual can also be given by

$$e_{(i)} = \frac{e_i}{1 - h_{ii}}$$

The PRESS statistic is defined as the sum of squares of the PRESS residuals and is therefore

$$\begin{aligned} PRESS &= \sum_{i=1}^{n} e_{(i)}^2 \\ &= \sum_{i=1}^{n} \left(\frac{e_i}{1 - h_{ii}} \right)^2 \end{aligned} \tag{8.13}$$

The PRESS statistic is a measure of how well the current model predicts each of the points in the data set if they were not included in the regression model, and is a good indication of the predictive power of the model. It is therefore desirable to minimise the PRESS statistic [68].

8.14 Residual Analysis

This section summarised from §2.3 – 2.6 of [68], and §2.7.1 of [4]

Residuals can be graphically explored to check the accuracy of the model. Many methods and approaches exist, so only a few of the most useful will be presented here.

During a regression analysis, several assumptions are made about the errors. These are that they are independent, have zero mean, have constant variance σ^2, and follow a normal distribution. The first two assumptions are usually summarised by $\epsilon_i \ N(0, \ \sigma^2)$, and the errors are assumed independent of one another. However, the estimation of the regression parameters β result in p normal equations, so the residuals (essentially the estimation of the errors) have $n-p$ degrees of freedom. So unless $n \gg p$ any non-normality checks will be compromised.

The histogram

A histogram can be used to check the normal distribution of the residuals. The residuals should be symmetrically distributed around zero with the general shape shown as in Fig. 8.7. These plots are rather crude, and it is sometimes difficult to judge when a distribution is "normal enough" to be indicative of an accurate model.

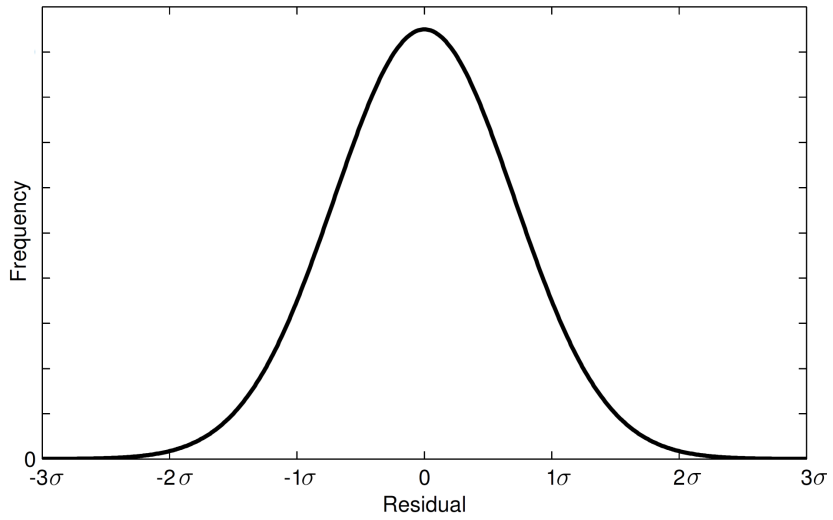


Figure 8.7: Shape of normal distribution for histogram plots

Normal probability plots

If the residuals are normally distributed the so-called *normal probability plot* shown in Fig. 8.8 will follow a straight line through the main middle bulk of the data. The data need not fall directly on the fitted line.

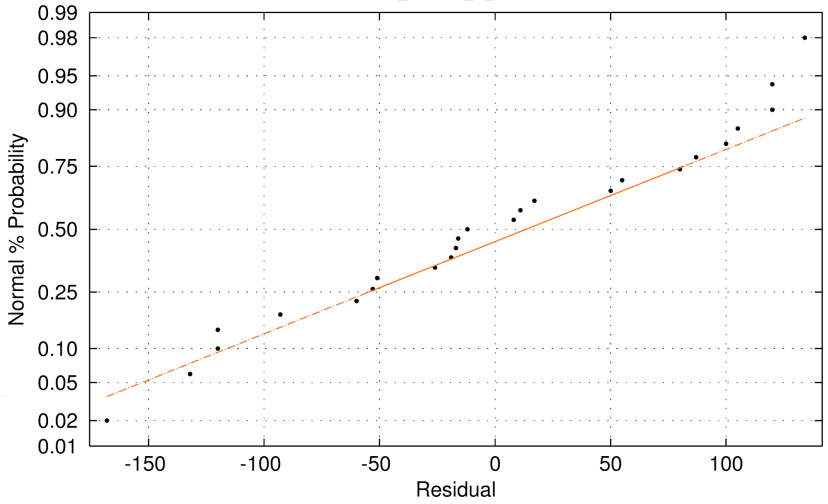


Figure 8.8: Normal probability plot example. (Plotted in Matlab from data in [68])

Plots of residual versus predicted response

Non-random patterns in these plots indicate model inadequacy. Outliers are characterised by any data points falling outside the range $-3\sigma < e_i < 3\sigma$, and indicate a poor fit in the region of the outlying data. It is therefore most useful to plot studentised residuals which are scaled to the standard deviation, as shown in Fig. 8.9 where the 3σ boundaries are included in the plot.

These plots will often exhibit a funnel-shaped pattern if the variance of the model depends on the mean of y. This suggests the need to transform the response

variable, as will be discussed in §8.16.

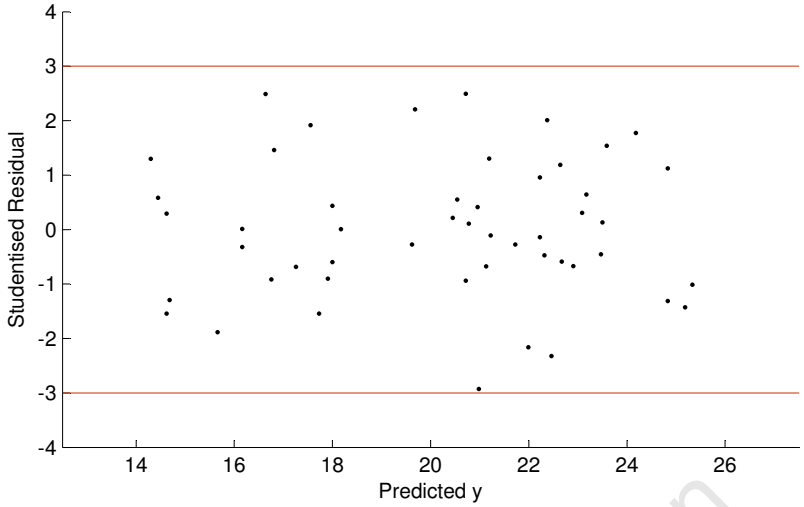


Figure 8.9: Example of a well-behaved residual scatter plot. Plotted in Matlab with data in [69] from [70].

8.15 Model Evaluation

This section summarised from [4]

There are several methods of quantitatively evaluating models, as will be discussed in the following sections.

8.15.1 The Multiple Correlation Coefficient, R^2

The use of the sum of squares in an ANOVA analysis helps us to evaluate different models. A better model will reduce the error sum of squares since the predicted responses will more closely match the actual response. The coefficient of multiple correlation R^2 is used to calculate this, where

$$R^2 = \frac{SS_r}{SS_t} = 1 - \frac{SS_e}{SS_t}$$

We can see that $0 \geq R^2 \geq 1$ and higher values of R^2 indicate a better model. However, addition of regressors to a model will always reduce the error sum of squares, increasing R^2 even when the additional regressor is not statistically significant. An adjusted R^2 statistic can be used which will decrease in value if unnecessary terms are added. This is given by

$$R^2_{adj} = 1 - \frac{SS_e/(n-p)}{SS_t/(n-1)} = 1 - \left(\frac{n-1}{n-p}\right)\left(1 - R^2\right)$$

8.15.2 The C_p Statistic

The C_p statistic is a measure of the total mean squared error for the p-term regression model. The total standardised mean squared error for a model with p terms is defined as

$$\begin{aligned}\Gamma_p &= \frac{1}{\sigma^2}\sum_{i=1}^{n}[E(y_i) - E(\hat{y}_i]^2 \\ &= \frac{1}{\sigma^2}\left[\text{bias}^2 + \text{variance}\right]\end{aligned}$$

C_p is an estimator of Γ_p and is given by

$$C_p = \frac{SS_e}{\hat{\sigma}^2} - n + 2p$$

If the model has negligible bias then

$$E(C_p|\text{zero bias}) = p$$

so C_p must be evaluated relative to p. If two potential models are being evaluated that have values of C_p equally close to their respective value of p, then the model with the smaller bias, and therefore smaller p, should be chosen.

8.15.3 Adequate Precision

Adequate precision is a measure of the range in predicted response relative to its associated error, and is therefore considered a signal to noise ratio. Its desired value is 4 or more.

$$\text{Adeq Precision} = \frac{\max(\widehat{y}) - \min(\widehat{y})}{\sqrt{\overline{V}(\widehat{y})}}$$

where

$$\overline{V}(\widehat{y}) = \frac{1}{n}\sum_{i=1}^{n} V\widehat{y} = \frac{p\sigma^2}{n}$$

8.16 Power Transformations on the Response

This section summarised from §6.7 of [4]

In order to model curvature a transformation of the response may be more successful than using a second-order response function. This will be most successful when the range on the response if fairly large; as a rule of thumb, when

$$\frac{y_{\max}}{y_{\min}} > 3$$

A procedure often used is the Box-Cox power transform. Here the response becomes y^λ, where $-2.5 < \lambda < 2.5$. Also, $\ln y$ is an important candidate model on the spectrum of λ, and formally the response takes the form

$$w = \frac{y^\lambda - 1}{\lambda}$$

In this way $\lambda = 0$ represents $\ln y$ as $\lim_{\lambda \to 0} w = \ln y$.

Although the procedure allows for a continuous spectrum of λ, "natural" choices should be used as listed in Table 8.9, where the values in the last column should be used for values of lambda calculated as given in the first column.

Table 8.9: Box-Cox power transforms

Best λ	**Equation**	**Name**	**Chosen λ**
1.5 – 2.5	$y^* = y^2$	Square	$\lambda = 2$
0.75 – 1.5	$y^* = y$	None	$\lambda = 1$
0.25 – 0.75	$y^* = y^{\frac{1}{2}}$	Square-root	$\lambda = 0.5$
-0.25 – 0.25	$y^* = \ln y$	Natural log	$\lambda = 0$
-0.75 – -0.25	$y^* = y^{-\frac{1}{2}}$	Inverse square-root	$\lambda = -0.5$
-1.5 – -0.75	$y^* = y^{-1}$	reciprocal	$\lambda = -1$
-2.5 – -1.5	$y^* = y^{-2}$	Inverse square	$\lambda = -2$

9 Optimisation Methodology

9.1 Introduction

This chapter will outline the method used to generate a mathematical model for optimisation of the airfoil. The optimisation problem will be discussed, where a comparison between using full-factorial and fractional-factorial arrays will be made.

The methods described here were formulated using aerodynamic results exclusively from the PABLO inviscid potential flow solutions. Matlab was used to assist setting up the fractional-factorial experiments, while statistical analyses were performed using the software package *Design Expert 7* by Stat-Ease[1].

9.2 Problem Description

9.2.1 The Merit Function

In order to carry out the optimisation procedure we need to formulate a merit function that represents our particular problem in a formal and mathematical way. This function is given generally by

$$M = \frac{\text{Effect}}{\text{Cost}} \tag{9.1}$$

We need to mathematically define the cost and effect, and then find a configuration that will minimise the cost while maximising the effect. We therefore need to maximise M.

The proposed configuration is for control to be achieved solely through activation of the tab, and to have the aileron float. The tab will therefore receive an applied moment, causing the aileron to rotate, resulting in a change of lift. The "cost" in equation 9.1 is then given by $\Delta C_{H_{tab}}$, and the "effect" by ΔC_L.

Activation of the tab will be achieved by a *smart memory alloy* (SMA), which deflects under the application of a current, but otherwise would not yield to aerodynamic forces [11]. The unactivated state would therefore leave $\delta_t = 0$, with the rudder at an angle depending on other geometric or freestream conditions. Activation of the SMA would deflect the tab by setting $\delta_t \neq 0$ which would then also change δ_a.

[1] Information regarding this package can be found at http://www.statease.com/expert.html

These two states of activation are shown in Fig. 9.1 for undeflected ($\delta_t = 0$) and deflected ($\delta_t \neq 0$) states of the tab. The energy cost would therefore be characterised by activation of the SMA tab, simply given by $C_{H_{t,\,\delta_t \neq 0}}$. The effect – the change in lift, ΔC_L – would need to be calculated from the change in lift between the two states of tab defection. This is most simply given by $C_{L_{\delta_t \neq 0}} - C_{L_{\delta_t = 0}}$. The merit function to be maximised is then given by

$$M = \frac{C_{L_{\delta_t \neq 0}} - C_{L_{\delta_t = 0}}}{C_{H_{t,\,\delta_t \neq 0}}} \tag{9.2}$$

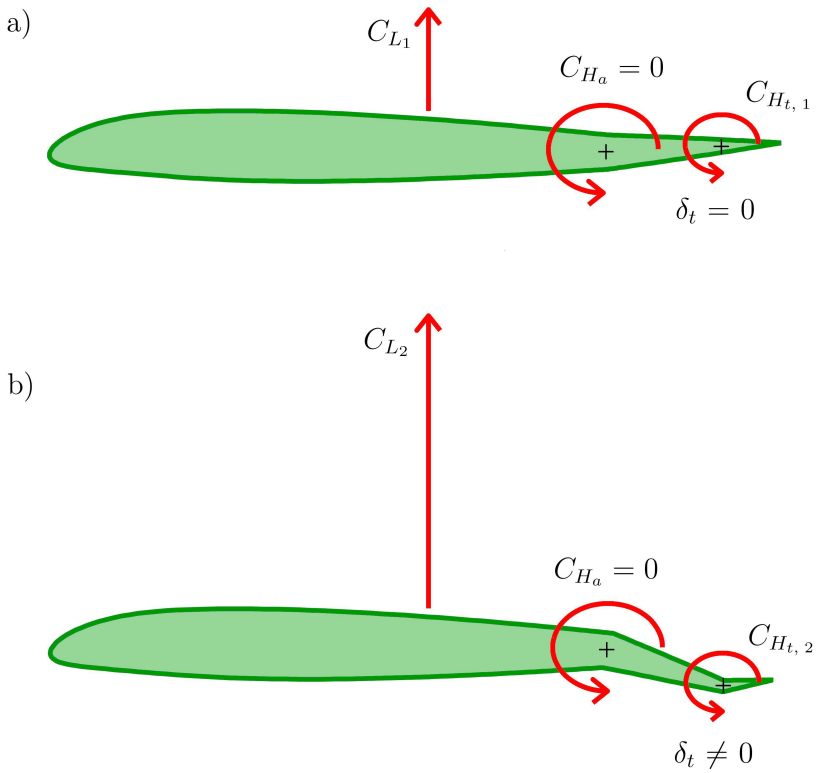


Figure 9.1: The two cases for the merit function: a floating aileron with a) $\delta_t = 0$, and b) $\delta_t \neq 0$.

In order to calculate the terms of the merit function so that a floating rudder is simulated, the aileron deflection must be calculated so that $C_{H_a} = 0$ for all configurations. However, the precise aileron deflection needed to achieve this state will not be known until a mathematical description of the system is obtained. The following sections detail steps taken to describe each response in terms of all other variables.

9.2.2 Factors and Responses

We need to express the responses in terms of each factor – or second-order combination of factors – that influences each response. The geometric configuration we want to optimise are the hinge locations as well as the camber and thickness

of the airfoil, and these need to be tested for a range of α and δ_t values to ensure the optimised geometry is valid over the full range of operating conditions.

The factors are listed in Table 9.1 along with the responses in which they need to be expressed, while the levels chosen are listed in Table 9.2. A detailed description of the levels chosen follows. For the numeric factors three levels were chosen so that curvature could be captured.

Table 9.1: Inputs and outputs of inviscid analysis program, as factors in the optimisation procedure.

Input	**Description**
α	angle of attack in degrees
δ_a	aileron deflection in degrees
δ_t	tab deflection in degrees
x_a	$\frac{x}{c}$-position of aileron hinge
x_t	$\frac{x}{c}$-position of tab hinge
NACA	NACA airfoil designation
Output	**Description**
C_L	Lift coefficient
C_{H_a}	Moment coefficient about the aileron hinge
C_{H_t}	Moment coefficient about the tab hinge

Table 9.2: Control factor levels for MDOE analysis

Control Factor	**Levels**		
α	-4°	0°	4°
δ_a	-4°	0°	4°
δ_t	-4°	0°	4°
x_a	0.74	0.76	0.78
x_t	0.9324	0.9424	0.9516
NACA	00012	23012	23018

angle of attack – α The range of α was chosen to ensure validity of the potential flow solution, which strictly applies to small angles only.

flap deflection angles – δ These angles were chosen to be smaller than the operating envelope to ensure validity of the potential flow solution, which strictly applies to small angles only. The levels were chosen specifically to reflect this. Angles quoted are in degrees, with positive clockwise to generate an increase in lift as is standard in aerospace applications.

Hinge locations – x The x-value of the aileron hinge location is expressed as a fraction of the total chord, with values arbitrarily chosen to cover a feasible range of values. With the existing A340 hinge located at approximately 76% of the chord, this was chosen as the mean level, with a 2% deviation for the maximum and minimum levels. The tab hinge was calculated in a similar way but using the percent of the aileron chord, then converted to a percentage of the total chord, by using the equation $x_t = 1 - (1 - x_a)^2$ to yield the values that appear in Table 9.2. These are dimensionless and represent the $\frac{x}{c}$ values.

Airfoil thickness/camber To simultaneously study the effect of thickness and camber with a single factor three airfoil profiles were chosen. Standard NACA 5-digit airfoil specifications were used – the symmetric 00012, and two similarly cambered airfoils of different thicknesses, the 23012 and 23018[2], the former of which has the same thickness as the symmetric airfoil.

Test arrays must now be selected to enable a comparison between the full-factorial and fractional-factorial analyses. The next section details the analysis and results for the full-factorial analysis. The fractional-factorial analysis and results will follow in §9.4.

[2] For more information on deriving the camber and thickness from the designation number refer to reference [71].

9.3 Full-Factorial Analysis

9.3.1 Setting Up The Experiment

A full factorial analysis was set up in Design Expert. Using three levels each for the six factors described above yielded a total of 729 configurations, which were run in Matlab to generate the three responses using PABLO. The test array and responses are output from Matlab into a Microsoft Excel spreadsheet for easy insertion into Design Expert. The code used is presented in Appendix F.1.

Before the data can be entered into Design Expert the experiment must first be set up. Figure 9.2 shows the first screen when selecting the "Historical Data" design option[3]. The numeric factors are defined here, with the categoric factor(s) on the following screen, shown in Fig. 9.3.

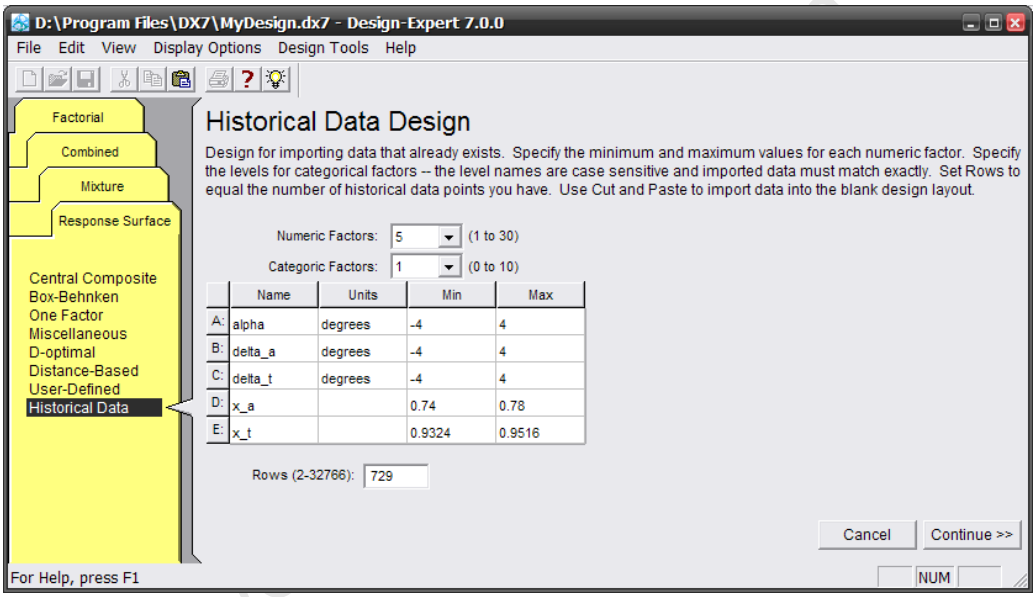


Figure 9.2: New design in Design Expert using the Historical Data option. The numeric factors are defined here.

The categoric factors in Fig. 9.3 can either be nominal or ordinal. From the Design Expert help files users are advised as such:

- Use nominal if the categoric levels are simply names or labels, such as Vendor A and Vendor B. In this case, it doesn't matter which one is first or second.
- Use ordinal if the levels represent a continuous relationship that is indicated by the order. An example of ordinal is low, medium and high or slow and fast.

The airfoil selection essentially combines the variation two different trends of camber and thickness, but cannot strictly be deemed to be of the ordinal type. This factor was therefore set to nominal.

[3] Although an option does exist for a full-factorial analysis, under General Factorial in the Factorial tab, it was found to be easier to use the Historical Data option. When performing the fractional-factorial analysis, a Taguchi OA option does exist, but the largest arrays available are L_{64}. The Historical Data option was therefore used throughout.

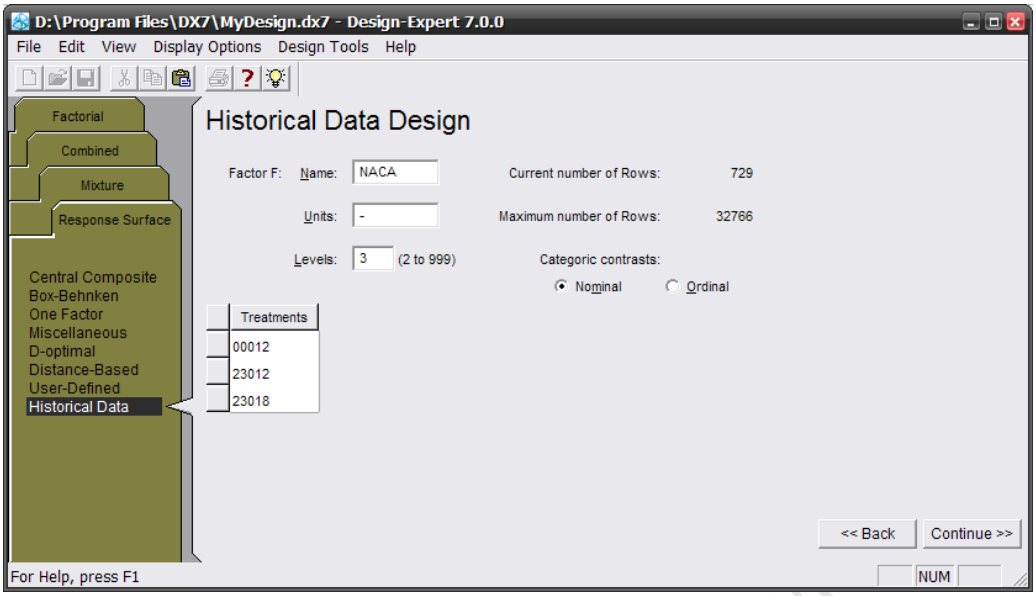


Figure 9.3: Definition of "categoric" factors in Design Expert

The last step in setting up the experiment requires definition of the responses. These are entered as shown in Fig. 9.4.

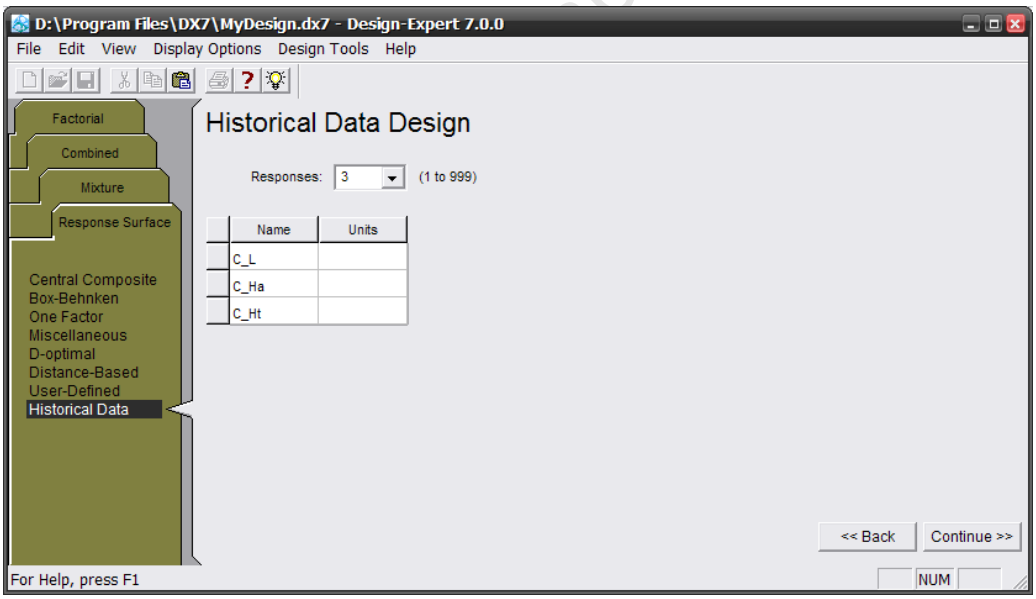
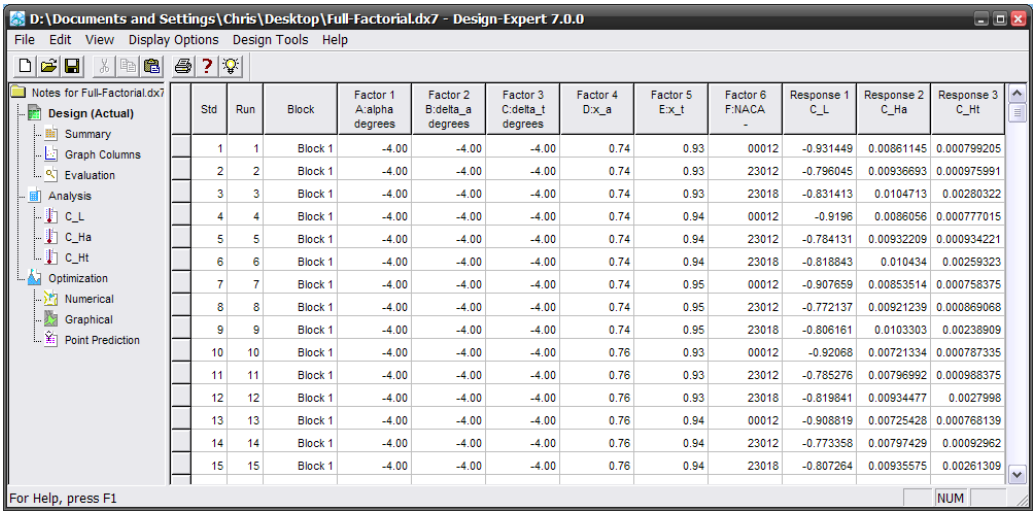


Figure 9.4: Definition of the responses in Design Expert

With the experiment set up the data can now be entered. Figure 9.5 shows the data inserted into the empty array created from the inputted factor definitions. The left pane shows the responses, which can be selected to initiate the actual analyses.

The goal here is to acquire a mathematical description of each response in terms of the factors. Selecting a response opens the Model tab, where the model and selection process is chosen, as shown in Fig. 9.6. The base model is chosen in the Process Order menu, where the Quadratic model has been chosen for this



Std	Run	Block	Factor 1 A:alpha degrees	Factor 2 B:delta_a degrees	Factor 3 C:delta_t degrees	Factor 4 D:x_a	Factor 5 E:x_t	Factor 6 F:NACA -	Response 1 C_L	Response 2 C_Ha	Response 3 C_Ht
1	1	Block 1	-4.00	-4.00	-4.00	0.74	0.93	00012	-0.931449	0.00861145	0.000799205
2	2	Block 1	-4.00	-4.00	-4.00	0.74	0.93	23012	-0.796045	0.00936693	0.000975991
3	3	Block 1	-4.00	-4.00	-4.00	0.74	0.93	23018	-0.831413	0.0104713	0.00280322
4	4	Block 1	-4.00	-4.00	-4.00	0.74	0.94	00012	-0.9196	0.0086056	0.000777015
5	5	Block 1	-4.00	-4.00	-4.00	0.74	0.94	23012	-0.784131	0.00932209	0.000934221
6	6	Block 1	-4.00	-4.00	-4.00	0.74	0.94	23018	-0.818843	0.010434	0.00259323
7	7	Block 1	-4.00	-4.00	-4.00	0.74	0.95	00012	-0.907659	0.00853514	0.000758375
8	8	Block 1	-4.00	-4.00	-4.00	0.74	0.95	23012	-0.772137	0.00921239	0.000869068
9	9	Block 1	-4.00	-4.00	-4.00	0.74	0.95	23018	-0.806161	0.0103303	0.00238909
10	10	Block 1	-4.00	-4.00	-4.00	0.76	0.93	00012	-0.92068	0.00721334	0.000787335
11	11	Block 1	-4.00	-4.00	-4.00	0.76	0.93	23012	-0.785276	0.00796992	0.000988375
12	12	Block 1	-4.00	-4.00	-4.00	0.76	0.93	23018	-0.819841	0.00934477	0.0027998
13	13	Block 1	-4.00	-4.00	-4.00	0.76	0.94	00012	-0.908819	0.00725428	0.000768139
14	14	Block 1	-4.00	-4.00	-4.00	0.76	0.94	23012	-0.773358	0.00797429	0.00092962
15	15	Block 1	-4.00	-4.00	-4.00	0.76	0.94	23018	-0.807264	0.00935575	0.00261309

Figure 9.5: Full-factorial data from Matlab inserted into Design Expert

example. This model selection causes certain terms in the list below to be selected for inclusion into the model, shown by the graphic M. Terms can also be manually included or excluded by a simple click on the chosen term to toggle the graphic.

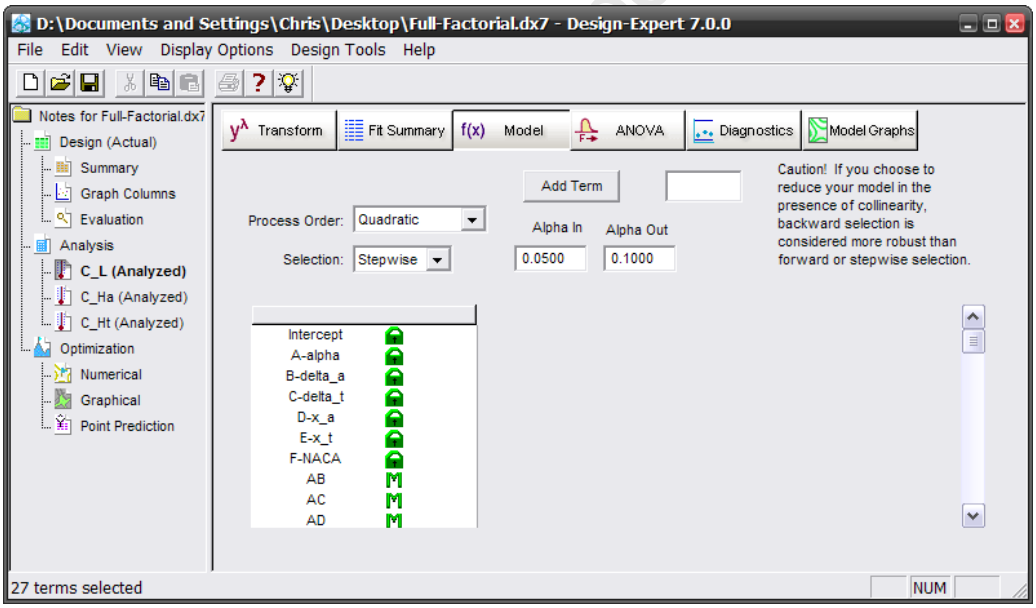


Figure 9.6: Model selection in Design Expert

A model refinement process can then be chosen in the Selection menu, where options are given for a backwards, forwards, or stepwise regression procedures. The Manual option in the Selection menu bypasses these. For backwards regression α_{out} can be set. For forwards regression α_{in} can be set. And for stepwise regression both α_{out} and α_{in} can be set. Here terms can be forcibly included, whether they are significant or not, and are signified by the graphic . If this is not done on first-order terms, if the model is found not to be hierarchical, an option is given to preserve hierarchy, as seen in Fig. 9.7. In one case a first-order term did not

appear in the final equations ($x_{h,\,a}$ in the L_{81} analysis on C_{H_t}), so it was chosen to preserve model hierarchy after the fact instead of forcing inclusion of terms from the onset.

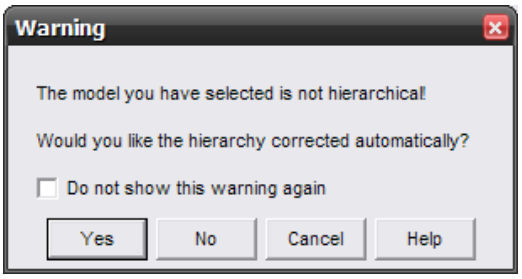


Figure 9.7: Dialog box to choose to preserve model hierarchy

9.3.2 Analysing The Output

Simply selecting the ANOVA tab initiates the analysis, and a large output results. This is shown (slightly edited of explanatory text but retaining all numerical output) on the following three pages for the example given above, where C_L is being analysed using stepwise regression to include quadratic terms, with $\alpha_{\text{in}} = 0.05$ and $\alpha_{\text{out}} = 0.1$.

Apart from a detailed ANOVA table showing all included terms, some model characteristics are given, as well as full equations in terms of coded and actual factors. The model characteristics are useful for comparing different models, and to this end the data was analysed using the following models: 2FI, Quadratic, Forwards, Backwards, and Stepwise.

For the data-driven term selection procedures the models were similar or the same. α_{in} and α_{out} were varied between the standard values of 0.1 and 0.05. For C_{Ha} and C_{H_t} the same terms were collected by each model for all settings of α. For C_L it was found that any setting of α_{in} or α_{out} to 0.05 would result in one model, and setting α exclusively to 0.1 would result in another, the difference being that a single extra term was found with α_{in} or α_{out} set to 0.05.

Tables 9.3 to 9.5 show model comparisons using the characteristics given in the ANOVA output shown at the bottom of Fig. 9.8. Here "Model A" refers to the model which included the $\alpha\, x_a$ interaction, and other than the extra term, the model differs from "Model B" only in a slight (0.7%) variation of the coefficient of the angle of attack, α, regressor. For comparative purposes the coefficient of variation (C.V.%) was dropped as it does not give meaningful results for responses that are close to zero [4].

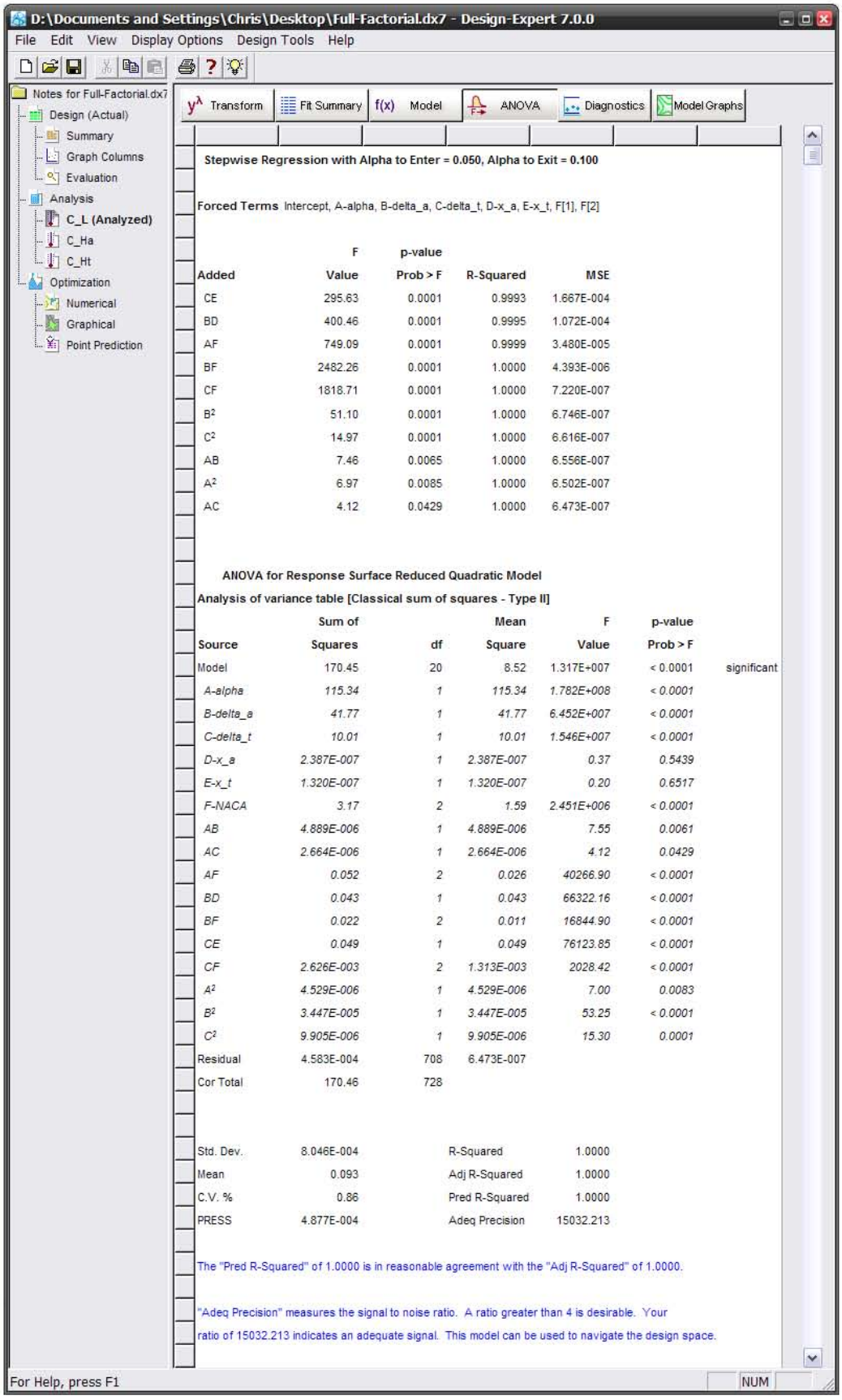


Figure 9.8: Page 1 of the Design Expert ANOVA output for the stepwise analysis of C_L to include all second-order terms

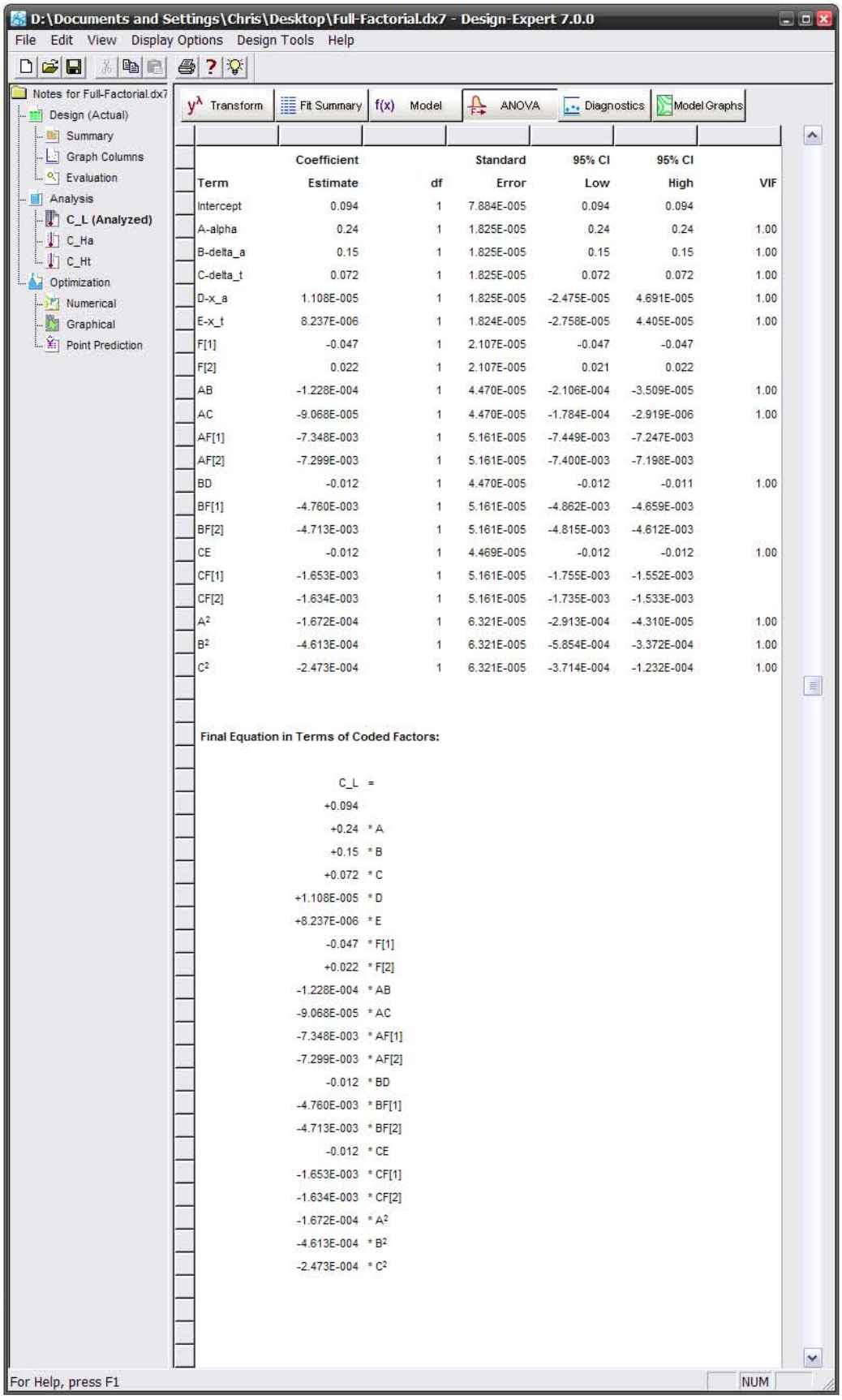


Figure 9.9: Page 2 of the Design Expert ANOVA output for the stepwise analysis of C_L to include all second-order terms

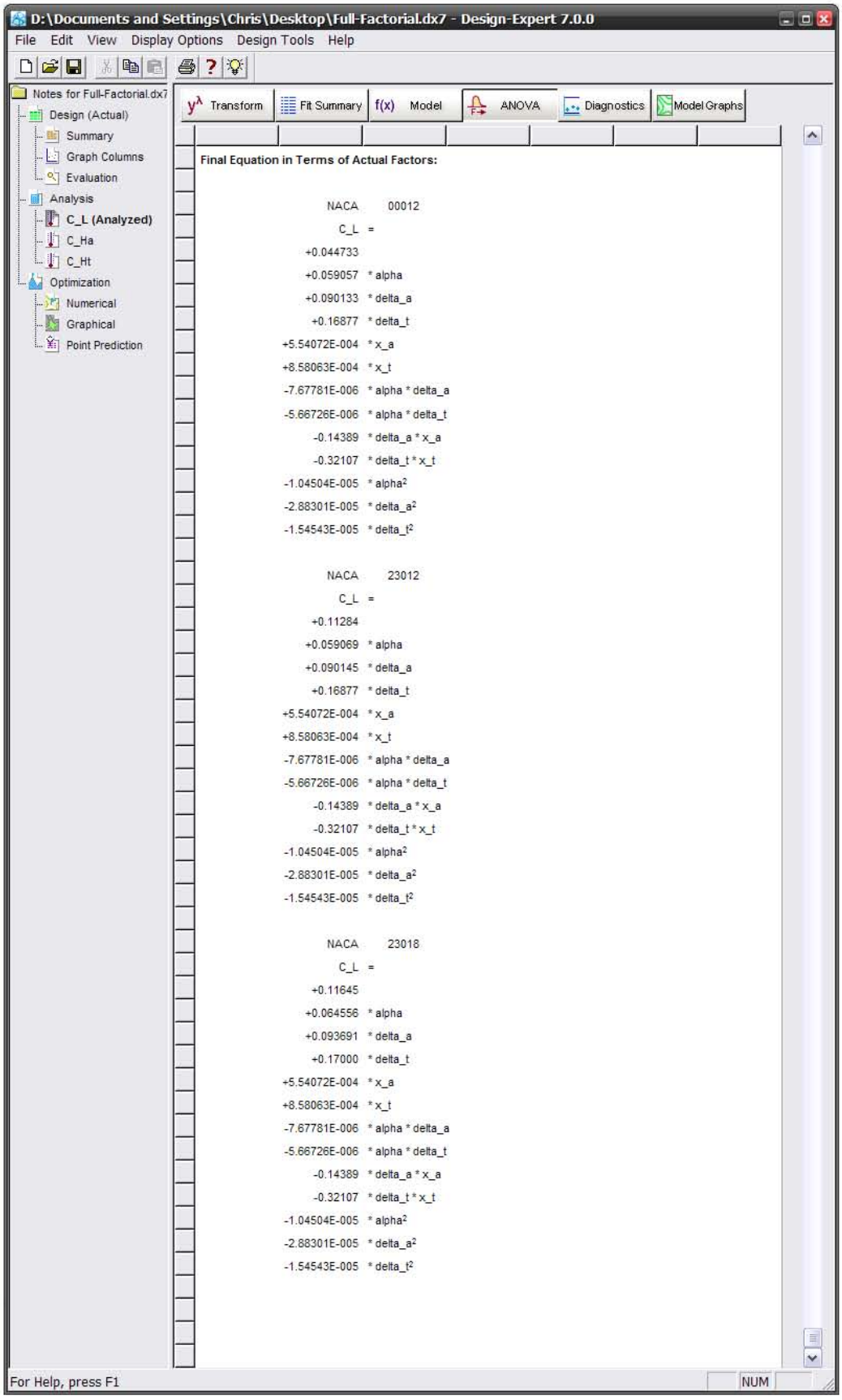


Figure 9.10: Page 3 of the Design Expert ANOVA output for the stepwise analysis of C_L to include all second-order terms

Table 9.3: Model characteristics for C_L full-factorial analysis

Characteristic	Model A	Model B	Quadratic	2FI
Std. Dev.	0.000803	0.000805	0.000809	0.000848
Mean	0.0932	0.0932	0.0932	0.0932
PRESS	0.000487	0.000488	0.000503	0.00055
R-Squared	0.99999732	0.99999731	0.99999733	1.00000
Adj R-Squared	0.99999725	0.99999724	0.99999721	0.9999969
Pred R-Squared	0.999997144	0.999997139	0.999997052	0.999996782
Adeq Precision	14712	15032	11931	12352

Table 9.4: Model characteristics for C_{H_a} full-factorial analysis

Characteristic	Reduced Model	Quadratic	2FI
Std. Dev.	0.000140	0.000141	0.000825
Mean	0.000553	0.000553	0.000553
PRESS	0.000015	0.000015	0.000511
R-Squared	0.999410	0.999411	0.97966
Adj R-Squared	0.999389	0.999383	0.97887
Pred R-Squared	0.999355	0.999341	0.97818
Adeq Precision	1045	941	175

Table 9.5: Model characteristics for C_{H_t} full-factorial analysis

Characteristic	Reduced Model	Quadratic	2FI
Std. Dev.	0.0000631	0.0000633	0.000170
Mean	0.001753	0.001753	0.001753
PRESS	0.0000031	0.0000031	0.000022
R-Squared	0.99544	0.99545	0.96698
Adj R-Squared	0.99528	0.99524	0.96571
Pred R-Squared	0.99499	0.99490	0.96437
Adeq Precision	309.6	279.4	113.0

For the models of C_{H_a} and C_{H_t} it's hardly surprising that the reduced regression models perform better than the standard models that were discussed in §8.11. Examination of Table 9.3 shows that for the "Model A" model, although the Adequate Precision value (a signal to noise ratio) is smaller, all other values, notably σ, PRESS, and R^2_{adj}, are all smaller where smaller is better. Although "Model A" was chosen for these reasons, the values are so close compared with the model with "Model B" that the difference is almost negligible. Examination of graphical methods of comparison were too similar to show any appreciable difference, as can be seen by comparing Figs. 9.11 to 9.14, starting overleaf. Final equations for each

response are presented in the following section.

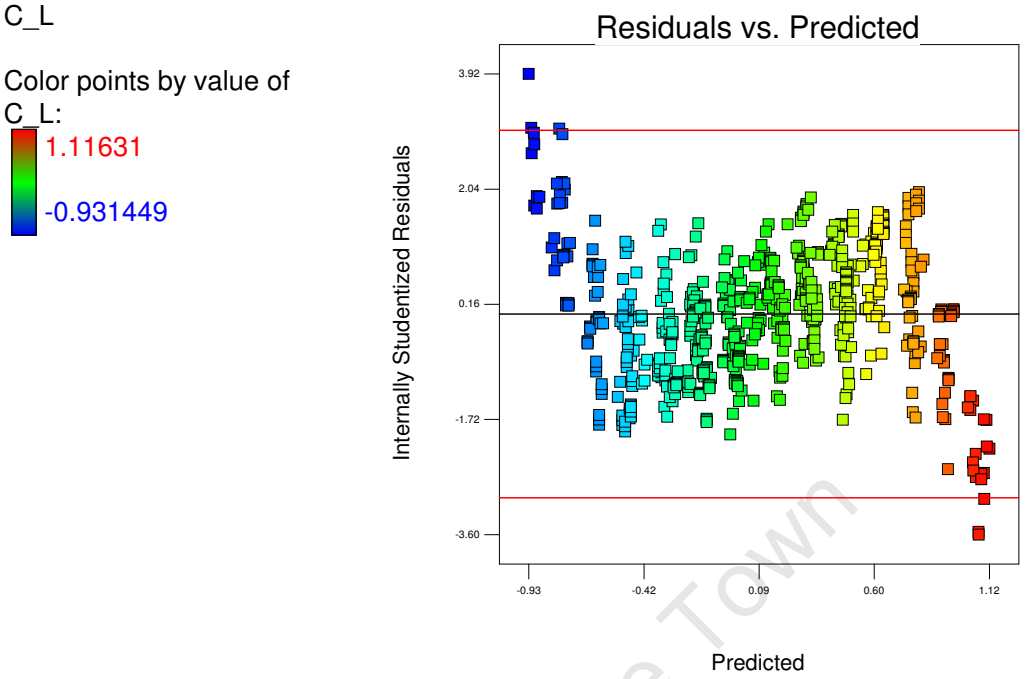


Figure 9.11: Design Expert normal plot of residuals for C_L regression models with α set to 0.1

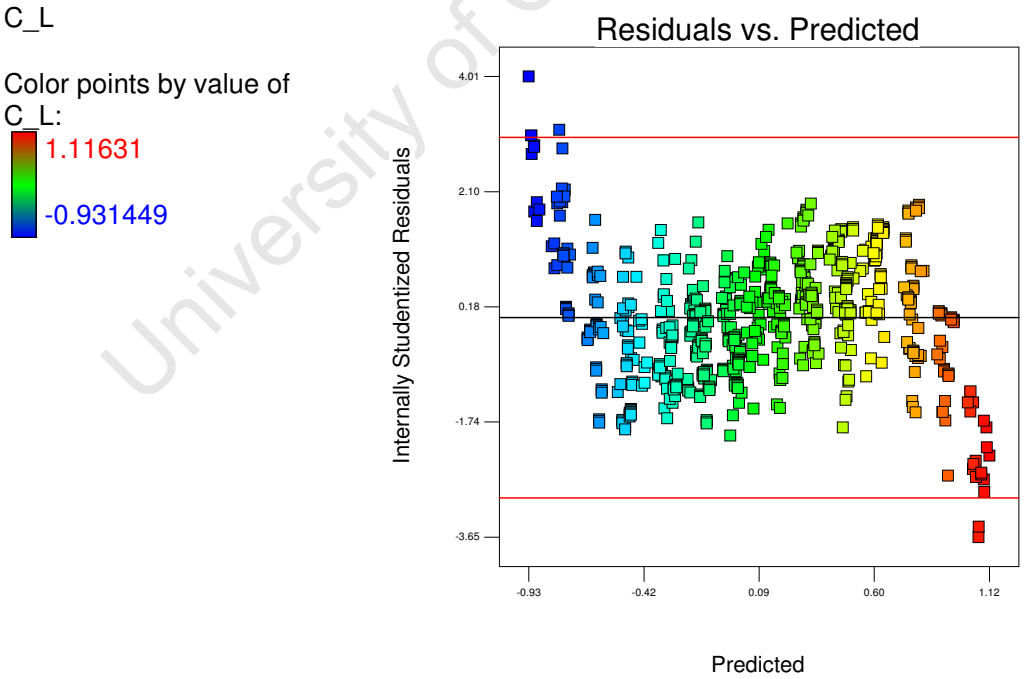


Figure 9.12: Design Expert normal plot of residuals for C_L regression models with α set to 0.05

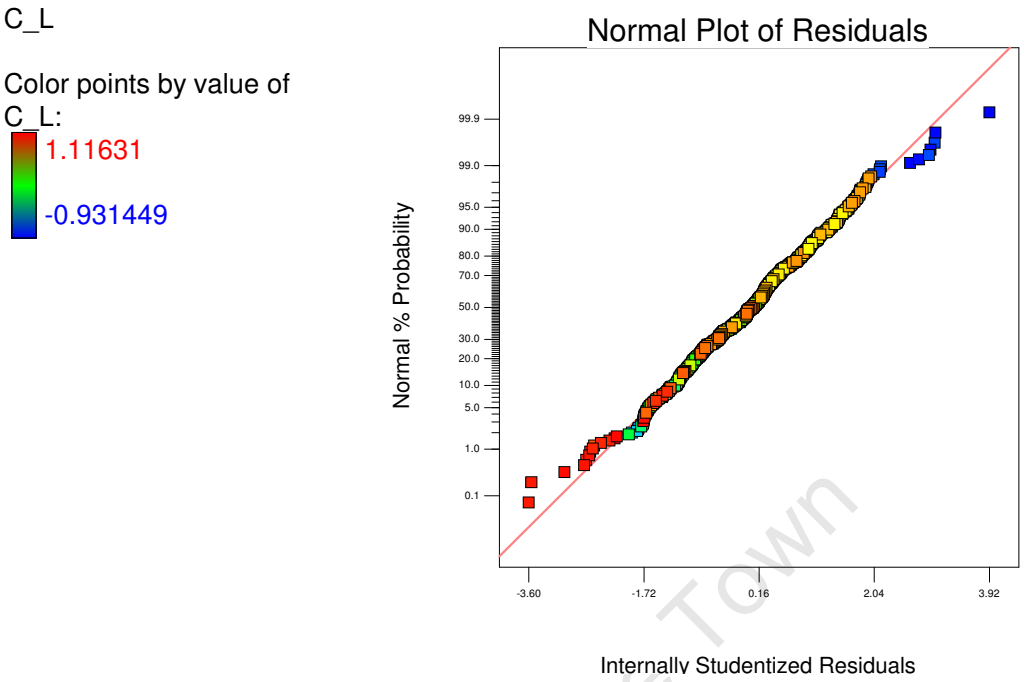


Figure 9.13: Design Expert plot of residual vs predicted values for C_L regression models with α set to 0.1

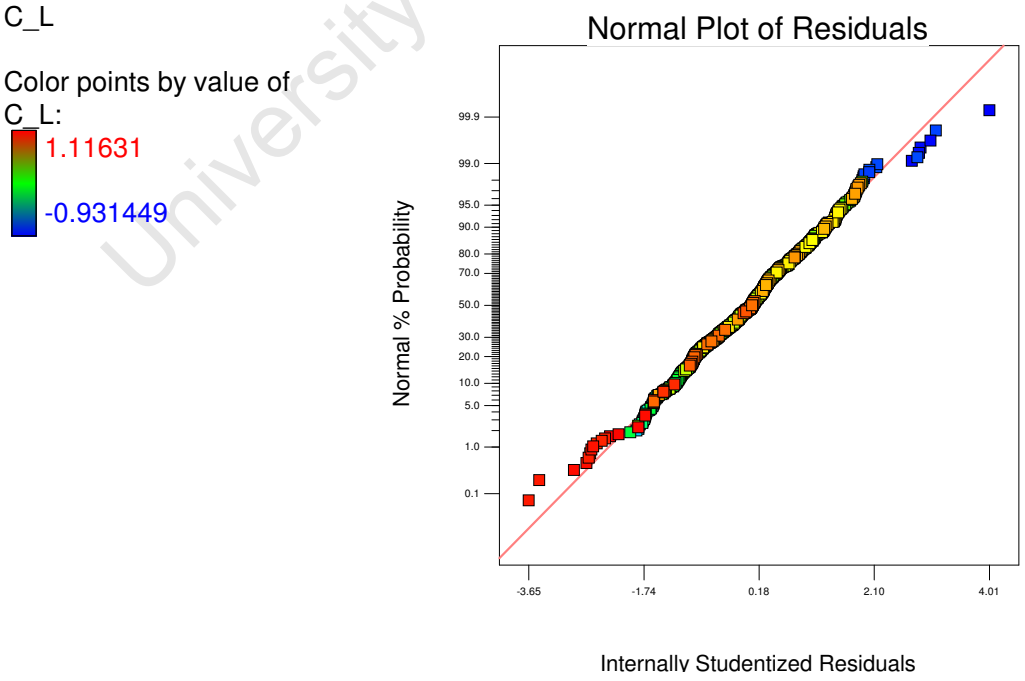


Figure 9.14: Design Expert plot of residual vs predicted values for C_L regression models with α set to 0.05

9.3.3 Resulting Equations

The following equations are in terms of the actual factors, as described in §9.2.2. Analysis of these equations will be left to Chapter 10 where a comparison between the two types of factorial analyses will be made. Equation parameters are as listed in Table 9.1.

$$\begin{aligned} C_{L_{00012}} &= -1.88 \times 10^{-03} + 1.19 \times 10^{-01}\alpha + 1.81 \times 10^{-01}\delta_a + 3.38 \times 10^{-01}\delta_t \\ &+ 1.11 \times 10^{-03}x_a + 1.72 \times 10^{-03}x_t - 7.68 \times 10^{-06}\alpha\,\delta_a - 5.67 \times 10^{-06}\alpha\,\delta_t \\ &+ 1.06 \times 10^{-03}\alpha\,x_a - 1.44 \times 10^{-01}\delta_a\,x_a - 3.21 \times 10^{-01}\delta_t\,x_t \\ &- 1.05 \times 10^{-05}\alpha^2 - 2.88 \times 10^{-05}\delta_a^2 - 1.55 \times 10^{-05}\delta_t^2 \end{aligned} \tag{9.3}$$

$$\begin{aligned} C_{L_{23012}} &= 1.34 \times 10^{-01} + 1.19 \times 10^{-01}\alpha + 1.81 \times 10^{-01}\delta_a + 3.38 \times 10^{-01}\delta_t \\ &+ 1.11 \times 10^{-03}x_a + 1.72 \times 10^{-03}x_t - 7.68 \times 10^{-06}\alpha\,\delta_a - 5.67 \times 10^{-06}\alpha\,\delta_t \\ &+ 1.06 \times 10^{-03}\alpha\,x_a - 1.44 \times 10^{-01}\delta_a\,x_a - 3.21 \times 10^{-01}\delta_t\,x_t \\ &- 1.05 \times 10^{-05}\alpha^2 - 2.88 \times 10^{-05}\delta_a^2 - 1.55 \times 10^{-05}\delta_t^2 \end{aligned} \tag{9.4}$$

$$\begin{aligned} C_{L_{23018}} &= 1.42 \times 10^{-01} + 1.25 \times 10^{-01}\alpha + 1.85 \times 10^{-01}\delta_a + 3.39 \times 10^{-01}\delta_t \\ &+ 1.11 \times 10^{-03}x_a + 1.72 \times 10^{-03}x_t - 7.68 \times 10^{-06}\alpha\,\delta_a - 5.67 \times 10^{-06}\alpha\,\delta_t \\ &+ 1.06 \times 10^{-03}\alpha\,x_a - 1.44 \times 10^{-01}\delta_a\,x_a - 3.21 \times 10^{-01}\delta_t\,x_t \\ &- 1.05 \times 10^{-05}\alpha^2 - 2.88 \times 10^{-05}\delta_a^2 - 1.55 \times 10^{-05}\delta_t^2 \end{aligned} \tag{9.5}$$

$$\begin{aligned} C_{H_{a,\ 00012}} &= -4.36 \times 10^{-02} - 4.53 \times 10^{-03}\alpha - 7.10 \times 10^{-03}\delta_a - 1.42 \times 10^{-02}\delta_t \\ &+ 9.52 \times 10^{-02}x_a + 4.50 \times 10^{-03}x_t - 4.39 \times 10^{-05}\alpha\,\delta_a - 6.33 \times 10^{-06}\alpha\,\delta_t \\ &+ 5.34 \times 10^{-03}\alpha\,x_a - 9.79 \times 10^{-05}\delta_a\,\delta_t + 8.23 \times 10^{-03}\delta_a\,x_a \\ &+ 9.46 \times 10^{-03}\delta_t\,x_a + 6.15 \times 10^{-03}\delta_t\,x_t + 9.25 \times 10^{-06}\alpha^2 \\ &- 1.03 \times 10^{-04}\delta_a^2 - 2.37 \times 10^{-05}\delta_t^2 - 5.44 \times 10^{-02}x_a^2 \end{aligned} \tag{9.6}$$

$$\begin{aligned} C_{H_{a,\ 23012}} &= -4.98 \times 10^{-02} - 4.57 \times 10^{-03}\alpha - 7.29 \times 10^{-03}\delta_a - 1.43 \times 10^{-02}\delta_t \\ &+ 1.02 \times 10^{-01}x_a + 4.50 \times 10^{-03}x_t - 4.39 \times 10^{-05}\alpha\,\delta_a - 6.33 \times 10^{-06}\alpha\,\delta_t \\ &+ 5.34 \times 10^{-03}\alpha\,x_a - 9.79 \times 10^{-05}\delta_a\,\delta_t + 8.23 \times 10^{-03}\delta_a\,x_a \\ &+ 9.46 \times 10^{-03}\delta_t\,x_a + 6.15 \times 10^{-03}\delta_t\,x_t + 9.25 \times 10^{-06}\alpha^2 \\ &- 1.03 \times 10^{-04}\delta_a^2 - 2.37 \times 10^{-05}\delta_t^2 - 5.44 \times 10^{-02}x_a^2 \end{aligned} \tag{9.7}$$

$$\begin{aligned} C_{H_{a,23018}} = &-5.31 \times 10^{-02} - 4.51 \times 10^{-03}\alpha - 7.22 \times 10^{-03}\delta_a - 1.42 \times 10^{-02}\delta_t \\ &+ 1.10 \times 10^{-01}x_a + 4.50 \times 10^{-03}x_t - 4.39 \times 10^{-05}\alpha\,\delta_a - 6.33 \times 10^{-06}\alpha\,\delta_t \\ &+ 5.34 \times 10^{-03}\alpha\,x_a - 9.79 \times 10^{-05}\delta_a\,\delta_t + 8.23 \times 10^{-03}\delta_a\,x_a \\ &+ 9.46 \times 10^{-03}\delta_t\,x_a + 6.15 \times 10^{-03}\delta_t\,x_t + 9.25 \times 10^{-06}\alpha^2 \\ &- 1.03 \times 10^{-04}\delta_a^2 - 2.37 \times 10^{-05}\delta_t^2 - 5.44 \times 10^{-02}x_a^2 \end{aligned} \tag{9.8}$$

$$\begin{aligned} C_{H_{t,\ 00012}} = &-1.01 \times 10^{-01} - 4.71 \times 10^{-04}\alpha - 8.24 \times 10^{-04}\delta_a - 1.77 \times 10^{-03}\delta_t \\ &- 1.21 \times 10^{-04}x_a + 2.24 \times 10^{-01}x_t - 1.88 \times 10^{-06}\alpha\,\delta_a \\ &- 3.86 \times 10^{-06}\alpha\,\delta_t + 4.88 \times 10^{-04}\alpha\,x_t - 2.75 \times 10^{-05}\delta_a\,\delta_t \\ &- 1.09 \times 10^{-04}\delta_a\,x_a + 9.52 \times 10^{-04}\delta_a\,x_t + 1.84 \times 10^{-03}\delta_t\,x_t \\ &+ 2.22 \times 10^{-06}\alpha^2 - 6.90 \times 10^{-06}\delta_a^2 - 1.92 \times 10^{-05}\delta_t^2 - 1.22 \times 10^{-01}x_t^2 \end{aligned} \tag{9.9}$$

$$\begin{aligned} C_{H_{t,\ 23012}} = &-1.01 \times 10^{-01} - 4.72 \times 10^{-04}\alpha - 8.36 \times 10^{-04}\delta_a - 1.80 \times 10^{-03}\delta_t \\ &- 1.21 \times 10^{-04}x_a + 2.25 \times 10^{-01}x_t - 1.88 \times 10^{-06}\alpha\,\delta_a \\ &- 3.86 \times 10^{-06}\alpha\,\delta_t + 4.88 \times 10^{-04}\alpha\,x_t - 2.75 \times 10^{-05}\delta_a\,\delta_t \\ &- 1.09 \times 10^{-04}\delta_a\,x_a + 9.52 \times 10^{-04}\delta_a\,x_t + 1.84 \times 10^{-03}\delta_t\,x_t \\ &+ 2.22 \times 10^{-06}\alpha^2 - 6.90 \times 10^{-06}\delta_a^2 - 1.92 \times 10^{-05}\delta_t^2 - 1.22 \times 10^{-01}x_t^2 \end{aligned} \tag{9.10}$$

$$\begin{aligned} C_{H_{t,23018}} = &-8.31 \times 10^{-02} - 4.68 \times 10^{-04}\alpha - 8.19 \times 10^{-04}\delta_a - 1.80 \times 10^{-03}\delta_t \\ &- 1.21 \times 10^{-04}x_a + 2.07 \times 10^{-01}x_t - 1.88 \times 10^{-06}\alpha\,\delta_a \\ &- 3.86 \times 10^{-06}\alpha\,\delta_t + 4.88 \times 10^{-04}\alpha\,x_t - 2.75 \times 10^{-05}\delta_a\,\delta_t \\ &- 1.09 \times 10^{-04}\delta_a\,x_a + 9.52 \times 10^{-04}\delta_a\,x_t + 1.84 \times 10^{-03}\delta_t\,x_t \\ &+ 2.22 \times 10^{-06}\alpha^2 - 6.90 \times 10^{-06}\delta_a^2 - 1.92 \times 10^{-05}\delta_t^2 - 1.22 \times 10^{-01}x_t^2 \end{aligned} \tag{9.11}$$

9.4 Fractional-Factorial Analysis

9.4.1 Selecting an Orthogonal Array

An array must be chosen for the fractional-factorial analysis. A choice is made with reference to the number and levels of factors and the degrees of freedom required. Assuming no prior knowledge of the interactions present, allowance must be made to check the significance of all two-factor interactions. Table 9.6 shows the calculation of the required degrees of freedom.

Table 9.6: Calculation of the degrees of freedom for the fractional-factorial analysis

Factor/Interaction	Degrees of freedom
Mean	1
Factors	$6 \times (3-1) = 12$
Interactions	$15 \times (3-1) \times (3-1) = 60$
Total	73

The table of available arrays in Appendix G is then consulted. The $L_{81}\,(3^{40})$ array has 80 degrees of freedom, and can accommodate only 3-level factors. It seems suitable, if confounding can be avoided when analysing all the interactions.

A series of files was written in Matlab, automating the process of generating the test matrix so that confounding is avoided, and calculating the responses for direct input to Design Expert. The Matlab code and explanations appear in Appendix F. Intermediate results will be described in the following sections.

9.4.2 Results of Factor Column Assignment

The factors were assigned to columns of a standard $L_{81}\,(3^{40})$ as listed in Table 9.7. Although the process of finding columns that were not confounded was automated, as a check the interaction table from [67] was consulted and Table 9.8 was compiled by hand.

Table 9.7: $L_{81}(3^{40})$ column assignment

Factor	Nomenclature	Column No.
α	A	1
δ_a	B	2
δ_t	C	5
x_a	D	9
x_t	E	14
NACA	F	18

It may be noticed that although factors are not in confounded columns, there are some repeated confounded columns (3, 6, 8, 15, 17, 27, 32). Special care was taken to ensure that this does not affect the result. Ultimate proof comes in the form of an example which can be found in Appendix H.

Table 9.8: Results of hand-checking with published interaction table

Interaction	**Columns used**		**Confounded columns**	
AB	1	2	3	4
AC	1	5	6	7
AD	1	9	8	10
AE	1	14	15	16
AF	1	18	17	19
BC	2	5	8	11
BD	2	9	6	12
BE	2	14	17	20
BF	2	18	15	21
CD	5	9	3	13
CE	5	14	23	32
CF	5	18	27	36
DE	9	14	27	40
DF	9	18	31	32
EF	14	18	3	22

9.4.3 Analysis in Design Expert

The factor levels and responses were entered into Design Expert as described for the full-factorial analysis. A similar model evaluation process was undertaken, with results shown in Tables 9.9 to 9.11.

Table 9.9: Model characteristics for C_L fractional-factorial analysis

Characteristic	**Model A**	**Model B**	**Quadratic**	**2FI**
Std. Dev.	0.00074	0.00076	0.00072	0.00077
Mean	0.0932	0.0932	0.0932	0.0932
PRESS	0.000061	0.000064	0.000073	0.000070
R-Squared	0.9999984	0.9999982	0.9999987	0.9999983
Adj R-Squared	0.9999977	0.9999976	0.9999978	0.9999975
Pred R-Squared	0.9999968	0.9999966	0.9999962	0.9999963
Adeq Precision	5062	5156	4431	4479

Table 9.10: Model characteristics for C_{H_a} fractional-factorial analysis

Characteristic	Reduced Model	Quadratic	2FI
Std. Dev.	0.00013	0.00014	0.00100
Mean	0.0006	0.0006	0.0006
PRESS	0.000003	0.000004	0.00013
R-Squared	0.99962	0.99963	0.97941
Adj R-Squared	0.99944	0.99938	0.96892
Pred R-Squared	0.9989	0.9984	0.9513
Adeq Precision	368	309	47

Table 9.11: Model characteristics for C_{H_a} fractional-factorial analysis

Characteristic	Reduced Model	Quadratic	2FI
Std. Dev.	0.000075	0.000078	0.000205
Mean	0.0018	0.0018	0.0018
PRESS	0.0000008	0.0000012	0.0000056
R-Squared	0.9951	0.9958	0.9674
Adj R-Squared	0.9935	0.9929	0.9508
Pred R-Squared	0.989	0.983	0.918
Adeq Precision	98	75	31

The results achieved were similar to those for the full-factorial analysis. In all cases the models with term selection procedures can be seen to perform better. (In the C_L analysis, although the full quadratic model has lower σ and higher R^2_{adj} values, the PRESS statistic and Adequate Precision are relatively poor.) The C_L analysis uncovered two different models for different settings of α_{in} and α_{out} (one with an additional x_a NACA interaction term), while C_{H_a} and C_{H_t} yielded only one model independent of the setting of α_{in} and α_{out}.

For the C_L analysis, the model labeled "Model A" in Table 9.9 was chosen. Here both models had the same terms, and the only difference was a slight variation in the value of the mean and the coefficient of $x_{h,\,a}$. Although the signal to noise ratio was poorer than "Model B", the PRESS statistic was better to a greater degree. Again examination of graphical methods of comparison were too similar to show any appreciable difference, as can be seen by Figs. 9.15 to 9.18, starting overleaf.

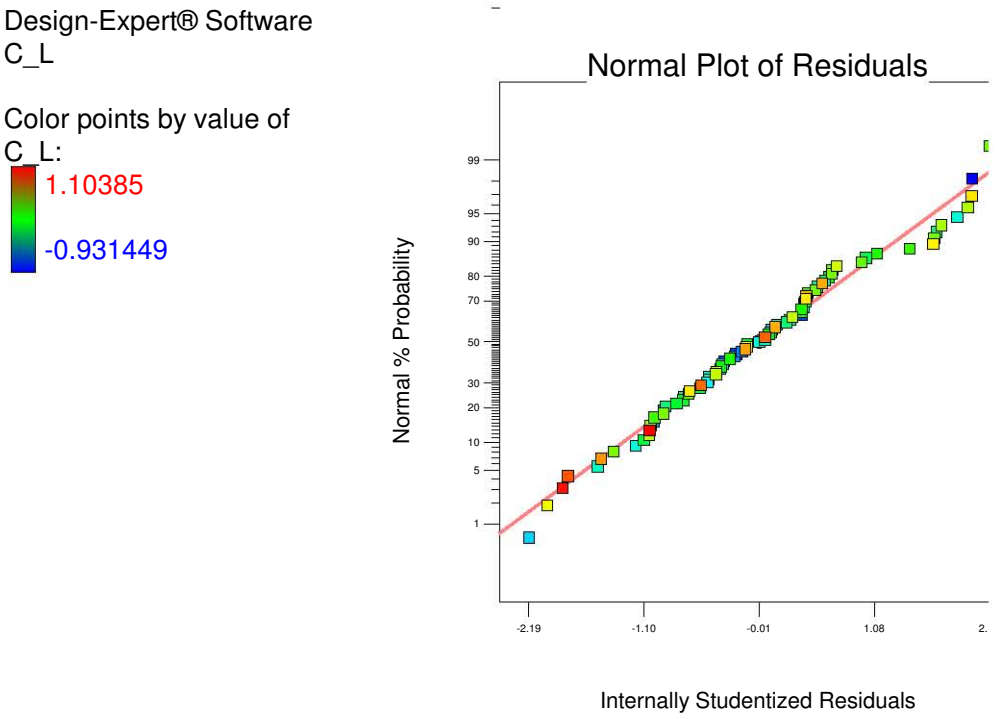


Figure 9.15: Design Expert normal plot of residuals for C_L regression models with α set to 0.1

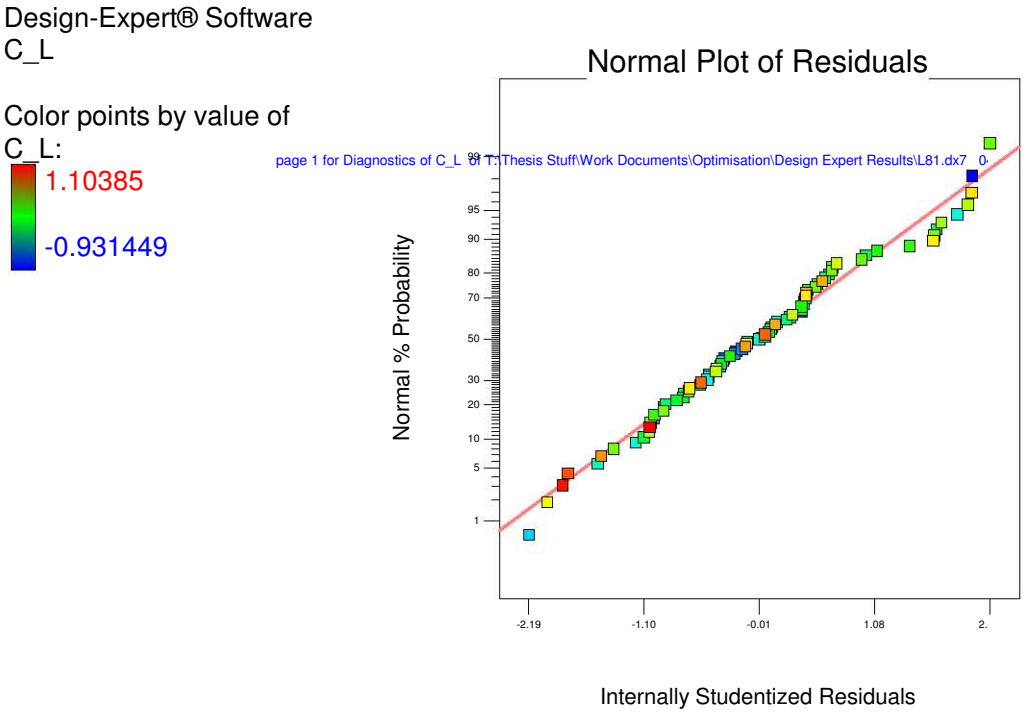


Figure 9.16: Design Expert normal plot of residuals for C_L regression models with α set to 0.05

page 1 for Diagnostics of C_L of T:\Thesis Stuff\Work Documents\Optimisation\Design Expert Results\L81.dx7 0

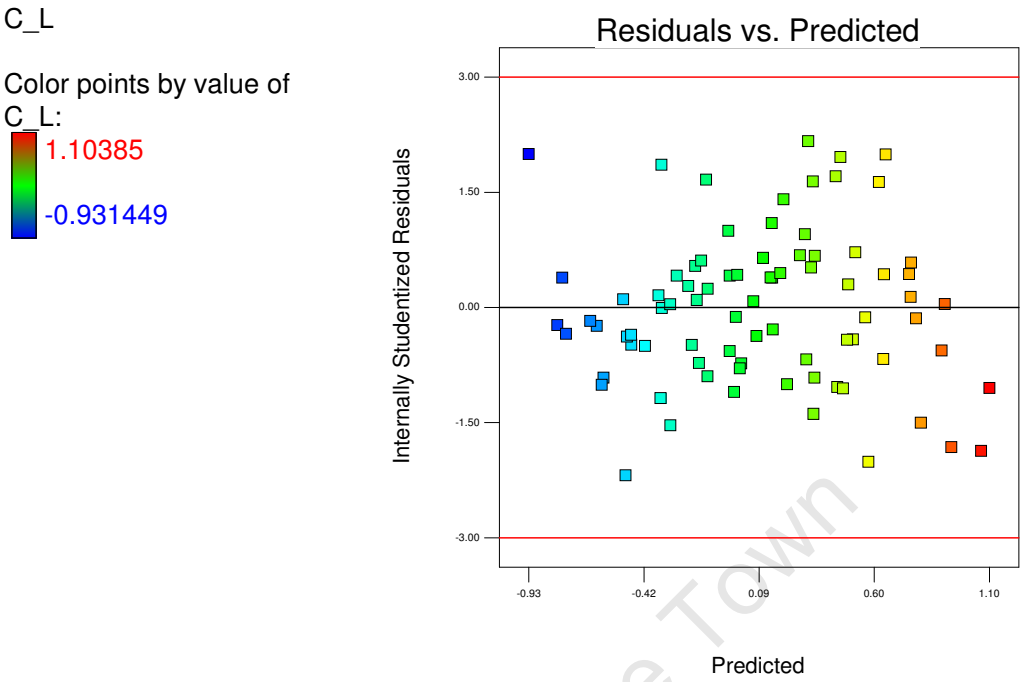


Figure 9.17: Design Expert plot of residual vs predicted values for C_L regression models with α set to 0.1

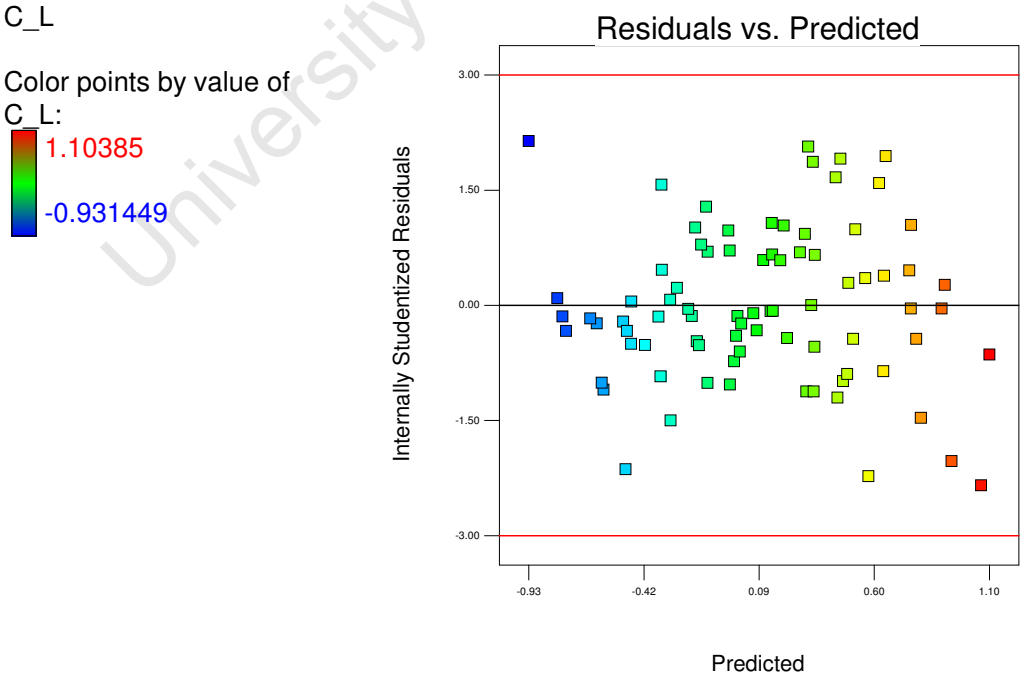


Figure 9.18: Design Expert plot of residual vs predicted values for C_L regression models with α set to 0.05

Final equations in terms of the actual factors appear in the following section, while analysis of the results continues in the following chapter, where factor effects will be studied and the merit function will be examined. Comparisons between the two models described here will be made.

9.4.4 Resulting Equations

The following equations are in terms of the actual factors, as described in §9.2.2. Analysis of these equations will be left to Chapter 10 where a comparison between the two types of factorial analyses will be made. Equation parameters are as listed in Table 9.1.

$$\begin{aligned} C_{L_{00012}} = {} & 4.02 \times 10^{-02} + 1.16 \times 10^{-01}\,\alpha + 1.78 \times 10^{-01}\,\delta_a + 3.32 \times 10^{-01}\,\delta_t \\ & - 1.19 \times 10^{-02}\,x_a - 3.27 \times 10^{-02}\,x_t - 1.68 \times 10^{-05}\,\alpha\,\delta_t \\ & + 4.78 \times 10^{-03}\,\alpha\,x_a - 1.40 \times 10^{-01}\,\delta_a\,x_a + 6.97 \times 10^{-03}\,\delta_t\,x_a \\ & - 3.20 \times 10^{-01}\,\delta_t\,x_t - 2.88 \times 10^{-05}\,\delta_a^2 \end{aligned} \tag{9.12}$$

$$\begin{aligned} C_{L_{23012}} = {} & 8.71 \times 10^{-02} 1.16 \times 10^{-01}\,\alpha + 1.78 \times 10^{-01}\,\delta_a + 3.32 \times 10^{-01}\,\delta_t \\ & + 9.30 \times 10^{-03}\,x_a + 4.49 \times 10^{-02}\,x_t - 1.68 \times 10^{-05}\,\alpha\,\delta_t \\ & + 4.78 \times 10^{-03}\,\alpha\,x_a - 1.40 \times 10^{-01}\,\delta_a\,x_a + 6.97 \times 10^{-03}\,\delta_t\,x_a \\ & - 3.20 \times 10^{-01}\,\delta_t\,x_t - 2.88 \times 10^{-05}\,\delta_a^2 \end{aligned} \tag{9.13}$$

$$\begin{aligned} C_{L_{23018}} = {} & 1.62 \times 10^{-01} 1.22 \times 10^{-01}\,\alpha 1.82 \times 10^{-01}\,\delta_a + 3.33 \times 10^{-01}\,\delta_t \\ & - 1.67 \times 10^{-02}\,x_a - 5.90 \times 10^{-03}\,x_t - 1.68 \times 10^{-05}\,\alpha\,\delta_t \\ & + 4.78 \times 10^{-03}\,\alpha\,x_a - 1.40 \times 10^{-01}\,\delta_a\,x_a + 6.97 \times 10^{-03}\,\delta_t\,x_a \\ & - 3.20 \times 10^{-01}\,\delta_t\,x_t - 2.88 \times 10^{-05}\,\delta_a^2 \end{aligned} \tag{9.14}$$

$$\begin{aligned} C_{H_{a,\ 00012}} = {} & -1.27 \times 10^{-02} - 4.75 \times 10^{-03}\,\alpha - 7.21 \times 10^{-03}\,\delta_a - 1.45 \times 10^{-02}\,\delta_t \\ & + 1.25 \times 10^{-02}\,x_a + 5.05 \times 10^{-03}\,x_t - 4.59 \times 10^{-05}\,\alpha\,\delta_a \\ & - 1.24 \times 10^{-05}\,\alpha\,\delta_t + 5.64 \times 10^{-03}\,\alpha\,x_a - 1.00 \times 10^{-04}\,\delta_a\,\delta_t \\ & + 8.37 \times 10^{-03}\,\delta_a\,x_a + 9.71 \times 10^{-03}\,\delta_t\,x_a + 6.32 \times 10^{-03}\,\delta_t\,x_t \\ & + 6.53 \times 10^{-06}\,\alpha^2 - 1.03 \times 10^{-04}\,\delta_a^2 - 2.69 \times 10^{-05}\,\delta_t^2 \end{aligned} \tag{9.15}$$

$$
\begin{aligned}
C_{H_{a,\ 23012}} = &-2.00 \times 10^{-02} - 4.79 \times 10^{-03}\,\alpha - 7.40 \times 10^{-03}\,\delta_a - 1.46 \times 10^{-02}\,\delta_t \\
&+ 2.13 \times 10^{-02}\,x_a + 5.05 \times 10^{-03}\,x_t - 4.59 \times 10^{-05}\,\alpha\,\delta_a \\
&- 1.24 \times 10^{-05}\,\alpha\,\delta_t + 5.64 \times 10^{-03}\,\alpha\,x_a - 1.00 \times 10^{-04}\,\delta_a\,\delta_t \\
&+ 8.37 \times 10^{-03}\,\delta_a\,x_a + 9.71 \times 10^{-03}\,\delta_t\,x_a + 6.32 \times 10^{-03}\,\delta_t\,x_t \\
&+ 6.53 \times 10^{-06}\,\alpha^2 - 1.03 \times 10^{-04}\,\delta_a^2 - 2.69 \times 10^{-05}\,\delta_t^2
\end{aligned}
\tag{9.16}
$$

$$
\begin{aligned}
C_{H_{a,\ 23018}} = &-2.17 \times 10^{-02} - 4.73 \times 10^{-03}\,\alpha - 7.33 \times 10^{-03}\,\delta_a - 1.46 \times 10^{-02}\,\delta_t \\
&+ 2.63 \times 10^{-02}\,x_a + 5.05 \times 10^{-03}\,x_t - 4.59 \times 10^{-05}\,\alpha\,\delta_a \\
&- 1.24 \times 10^{-05}\,\alpha\,\delta_t + 5.64 \times 10^{-03}\,\alpha\,x_a - 1.00 \times 10^{-04}\,\delta_a\,\delta_t \\
&+ 8.37 \times 10^{-03}\,\delta_a\,x_a + 9.71 \times 10^{-03}\,\delta_t\,x_a + 6.32 \times 10^{-03}\,\delta_t\,x_t \\
&+ 6.53 \times 10^{-06}\,\alpha^2 - 1.03 \times 10^{-04}\,\delta_a^2 - 2.69 \times 10^{-05}\,\delta_t^2
\end{aligned}
\tag{9.17}
$$

$$
\begin{aligned}
C_{H_{t,\ 00012}} = &\ 7.23 \times 10^{-03} - 9.97 \times 10^{-06}\,\alpha - 1.00 \times 10^{-03}\,\delta_a - 1.77 \times 10^{-03}\,\delta_t \\
&- 6.15 \times 10^{-03}\,x_t - 1.98 \times 10^{-06}\,\alpha\,\delta_a - 3.73 \times 10^{-06}\,\alpha\,\delta_t \\
&- 2.74 \times 10^{-05}\,\delta_a\,\delta_t + 1.06 \times 10^{-03}\,\delta_a\,x_t + 1.83 \times 10^{-03}\,\delta_t\,x_t \\
&+ 2.38 \times 10^{-06}\,\alpha^2 - 6.85 \times 10^{-06}\,\delta_a^2 - 1.92 \times 10^{-05}\,\delta_t^2
\end{aligned}
\tag{9.18}
$$

$$
\begin{aligned}
C_{H_{t,\ 23012}} = &\ 7.55 \times 10^{-03} - 9.97 \times 10^{-06}\,\alpha - 1.02 \times 10^{-03}\,\delta_a - 1.80 \times 10^{-03}\,\delta_t \\
&- 6.51 \times 10^{-03}\,x_t - 1.98 \times 10^{-06}\,\alpha\,\delta_a - 3.73 \times 10^{-06}\,\alpha\,\delta_t \\
&- 2.74 \times 10^{-05}\,\delta_a\,\delta_t + 1.06 \times 10^{-03}\,\delta_a\,x_t + 1.83 \times 10^{-03}\,\delta_t\,x_t \\
&+ 2.38 \times 10^{-06}\,\alpha^2 - 6.85 \times 10^{-06}\,\delta_a^2 - 1.92 \times 10^{-05}\,\delta_t^2
\end{aligned}
\tag{9.19}
$$

$$
\begin{aligned}
C_{H_{t,\ 23018}} = &\ 2.48 \times 10^{-02} - 9.97 \times 10^{-06}\,\alpha - 9.96 \times 10^{-04}\,\delta_a - 1.79 \times 10^{-03}\,\delta_t \\
&- 2.29 \times 10^{-02}\,x_t - 1.98 \times 10^{-06}\,\alpha\,\delta_a - 3.73 \times 10^{-06}\,\alpha\,\delta_t \\
&- 2.74 \times 10^{-05}\,\delta_a\,\delta_t + 1.06 \times 10^{-03}\,\delta_a\,x_t + 1.83 \times 10^{-03}\,\delta_t\,x_t \\
&+ 2.38 \times 10^{-06}\,\alpha^2 - 6.85 \times 10^{-06}\,\delta_a^2 - 1.92 \times 10^{-05}\,\delta_t^2
\end{aligned}
\tag{9.20}
$$

10 Final Analysis And Results

The purpose of this chapter is two-fold. A comparison must be made between the results using each of the two methods outlined in the previous chapter, while an analysis of the results with a view to optimising the airfoil must also be undertaken.

First the model terms and their coefficients will be compared. Then factor effects will be examined and compared. Finally the merit function will be examined.

10.1 Model Terms And Coefficients

Tables 10.1 to 10.3 contain a list of model terms and coefficients for each reponse, enabling comparison between the full- and fractional-factorial analyses. The equations are in terms of coded variables to allow for direct comparison of all terms including the categoric factor *NACA*. Scientific notation has been used, except where coefficients differ by an order of magnitude. Where coefficients are the same order of magnitude, they are reported to at least 3 significant figures, or so that a difference of more than 1 unit in the least significant figure is seen. The coefficients were taken from the equations for each response in terms of the coded variables. Use of coded variables allows comparison of the relative significance of each term, with larger coefficients being more significant.

The difference between coefficients are reported in the "% Error" column, where the error is reported as the percentage difference between the two values as a fraction of the full-factorial value.

Some of the least significant terms of the full-factorial analysis were not picked up in the fractional-factorial analysis, while some additional terms were picked up. The error is generally small, but can be quite large, especially for the least significant terms (i.e. those with small coefficients).

Table 10.1: Comparison of model terms and coefficients for C_L analysis

Term	**Factor(s)**	**Full-factorial**	**L_{81}**	**% Error**
Intercept		9.38×10^{-02}	9.35×10^{-02}	0.29%
A	α	4.87151×10^{-01}	4.87142×10^{-01}	0.0018%
B	δ_a	2.93159×10^{-01}	2.93156×10^{-01}	0.0012%
C	δ_t	1.4368×10^{-01}	1.4365×10^{-01}	0.022%
D	x_a	0.00002	-0.00013	682%
E	x_t	1.65×10^{-05}	2.03×10^{-05}	23%
F[1]	NACA	-9.322×10^{-02}	-9.320×10^{-02}	0.016%
F[2]	NACA	4.3003×10^{-02}	4.2991×10^{-02}	0.028%
AB	$\alpha\,\delta_a$	-1.23×10^{-04}		
AC	$\alpha\,\delta_t$	-0.00009	-0.00027	196%
AD	$\alpha\,x_a$	0.00008	0.00038	351%
AE	$\alpha\,x_t$			
AF[1]	α NACA	-7.348×10^{-03}	-7.343×10^{-03}	0.073%
AF[2]	α NACA	-7.299×10^{-03}	-7.304×10^{-03}	0.077%
BC	$\delta_a\,\delta_t$			
BD	$\delta_a\,x_a$	-1.15×10^{-02}	-1.12×10^{-02}	2.9%
BE	$\delta_a\,x_t$			
BF[1]	δ_a NACA	-4.7602×10^{-03}	-4.7606×10^{-03}	0.0077%
BF[2]	δ_a NACA	-4.71×10^{-03}	-4.74×10^{-03}	0.61%
CD	$\delta_t\,x_a$		5.57×10^{-04}	
CE	$\delta_t\,x_t$	-1.233×10^{-02}	-1.230×10^{-02}	0.20%
CF[1]	δ_t NACA	-1.653×10^{-03}	-1.651×10^{-03}	0.16%
CF[2]	δ_t NACA	-1.634×10^{-03}	-1.642×10^{-03}	0.49%
DE	$x_a\,x_t$			
DF[1]	x_a NACA		-1.09×10^{-04}	
DF[2]	x_a NACA		3.15×10^{-04}	
EF[1]	x_t NACA		-3.34×10^{-04}	
EF[2]	x_t NACA		4.11×10^{-04}	
A^2	α^2	-1.67×10^{-04}		
B^2	δ_a^2	-4.613×10^{-04}	-4.611×10^{-04}	0.042%
C^2	δ_t^2	-2.47×10^{-04}		
D^2	x_a^2			
E^2	x_t^2			
F^2	$NACA^2$			

Table 10.2: Comparison of model terms and coefficients for C_{H_a} analysis

Term	**Factor(s)**	**Full-factorial**	**L_{81}**	**% Error**
Intercept		1.81×10^{-03}	1.86×10^{-03}	2.7%
A	α	-1.91×10^{-03}	-1.89×10^{-03}	0.87%
B	δ_a	-3.789×10^{-03}	-3.787×10^{-03}	0.052%
C	δ_t	-4.99×10^{-03}	-4.97×10^{-03}	0.37%
D	x_a	3.95×10^{-04}	4.01×10^{-04}	1.5%
E	x_t	4.32×10^{-05}	4.84×10^{-05}	12%
F[1]	NACA	-2.73×10^{-04}	-2.53×10^{-04}	7.3%
F[2]	NACA	-9.49×10^{-04}	-9.48×10^{-04}	0.17%
AB	$\alpha\,\delta_a$	-7.02×10^{-04}	-7.35×10^{-04}	4.7%
AC	$\alpha\,\delta_t$	-1.01×10^{-04}	-1.99×10^{-04}	97%
AD	$\alpha\,x_a$	4.27×10^{-04}	4.51×10^{-04}	5.6%
AE	$\alpha\,x_t$			
AF[1]	α NACA	3.2611×10^{-05}	3.2605×10^{-05}	0.019%
AF[2]	α NACA	-1.34×10^{-04}	-1.40×10^{-04}	4.2%
BC	$\delta_a\,\delta_t$	-1.57×10^{-03}	-1.60×10^{-03}	2.4%
BD	$\delta_a\,x_a$	6.58×10^{-04}	6.70×10^{-04}	1.8%
BE	$\delta_a\,x_t$			
BF[1]	δ_a NACA	4.13×10^{-04}	4.16×10^{-04}	0.59%
BF[2]	δ_a NACA	-3.4299×10^{-04}	-3.4292×10^{-04}	0.021%
CD	$\delta_t\,x_a$	7.57×10^{-04}	7.77×10^{-04}	2.6%
CE	$\delta_t\,x_t$	2.36×10^{-04}	2.43×10^{-04}	2.7%
CF[1]	δ_t NACA	1.93×10^{-04}	1.98×10^{-04}	2.8%
CF[2]	δ_t NACA	-2.54×10^{-04}	-2.68×10^{-04}	5.6%
DE	$x_a\,x_t$			
DF[1]	x_a NACA	-1.44×10^{-04}	-1.50×10^{-04}	4.3%
DF[2]	x_a NACA	-0.0000004	0.0000243	5832%
EF[1]	x_t NACA			
EF[2]	x_t NACA			
A^2	α^2	1.48×10^{-04}	1.04×10^{-04}	29%
B^2	δ_a^2	-1.6406×10^{-03}	-1.6411×10^{-03}	0.029%
C^2	δ_t^2	-3.79×10^{-04}	-4.31×10^{-04}	14%
D^2	x_a^2	-2.17×10^{-05}		
E^2	x_t^2			
F^2	$NACA^2$			

Table 10.3: Comparison of model terms and coefficients for C_{H_t} analysis

Term	**Factor(s)**	**Full-factorial**	**L_{81}**	**% Error**
Intercept		2.017×10^{-03}	2.008×10^{-03}	0.45%
A	α	-4.09×10^{-05}	-3.99×10^{-05}	2.6%
B	δ_a	-4.89×10^{-05}	-4.91×10^{-05}	0.27%
C	δ_t	-2.424×10^{-04}	-2.429×10^{-04}	0.18%
D	x_a	-2.42×10^{-06}		
E	x_t	-1.17×10^{-04}	-1.14×10^{-04}	2.3%
F[1]	NACA	-5.661×10^{-04}	-5.652×10^{-04}	0.17%
F[2]	NACA	-5.89×10^{-04}	-5.94×10^{-04}	0.81%
AB	$\alpha\,\delta_a$	-3.00×10^{-05}	-3.16×10^{-05}	5.4%
AC	$\alpha\,\delta_t$	-6.17×10^{-05}	-5.97×10^{-05}	3.2%
AD	$\alpha\,x_a$			
AE	$\alpha\,x_t$	1.88×10^{-05}		
AF[1]	α NACA	-3.26×10^{-06}		
AF[2]	α NACA	-7.08×10^{-06}		
BC	$\delta_a\,\delta_t$	-4.394×10^{-04}	-4.386×10^{-04}	0.17%
BD	$\delta_a\,x_a$	-0.000009	0.000041	565%
BE	$\delta_a\,x_t$	0.000037	0.000009	75%
BF[1]	δ_a NACA	0.000008	-0.000052	726%
BF[2]	δ_a NACA	-3.99×10^{-05}		
CD	$\delta_t\,x_a$			
CE	$\delta_t\,x_t$	7.054×10^{-05}	7.044×10^{-05}	0.14%
CF[1]	δ_t NACA	7.20×10^{-05}	7.03×10^{-05}	2.3%
CF[2]	δ_t NACA	-5.327×10^{-05}	-5.336×10^{-05}	0.16%
DE	$x_a\,x_t$			
DF[1]	x_a NACA			
DF[2]	x_a NACA			
EF[1]	x_t NACA	5.15×10^{-05}	5.49×10^{-05}	6.7%
EF[2]	x_t NACA	5.84×10^{-05}	5.14×10^{-05}	12%
A^2	α^2	3.56×10^{-05}	3.81×10^{-05}	7.0%
B^2	δ_a^2	-1.103×10^{-04}	-1.096×10^{-04}	0.71%
C^2	δ_t^2	-3.070×10^{-04}	-3.073×10^{-04}	0.10%
D^2	x_a^2			
E^2	x_t^2	-1.13×10^{-05}		
F^2	$NACA^2$			

10.2 Factor Effects

The factor effects are calculated from the raw data of the responses averaged over the levels of each factor, so are independent of the mathematical models calculated in the previous section. It can be seen from Figs. 10.1 to 10.6 that the average values calculated are quite similar. Tables are therefore produced for a numerical comparison as well as percentage differences for each level of each factor for the three responses. The percentage difference, reported below each figure pair, is calculated as per the full-factorial analysis. i.e.

$$\% \text{ Error} = \text{abs}\left(\frac{\overline{F}_{i\,\text{full}} - \overline{F}_{i\,\text{frac}}}{\overline{F}_{i\,\text{full}}}\right) \times 100\%$$

The actual average values for each response are shown grouped on page 158. Factor levels in Figs. 10.1 to 10.6 are as in Table 9.2, with the lower level shown to the left.

It can be seen that the factor effects are very similar for both analyses used even though the fractional-factorial analysis has only 11% of the data points compared with the fractional factorial analysis.

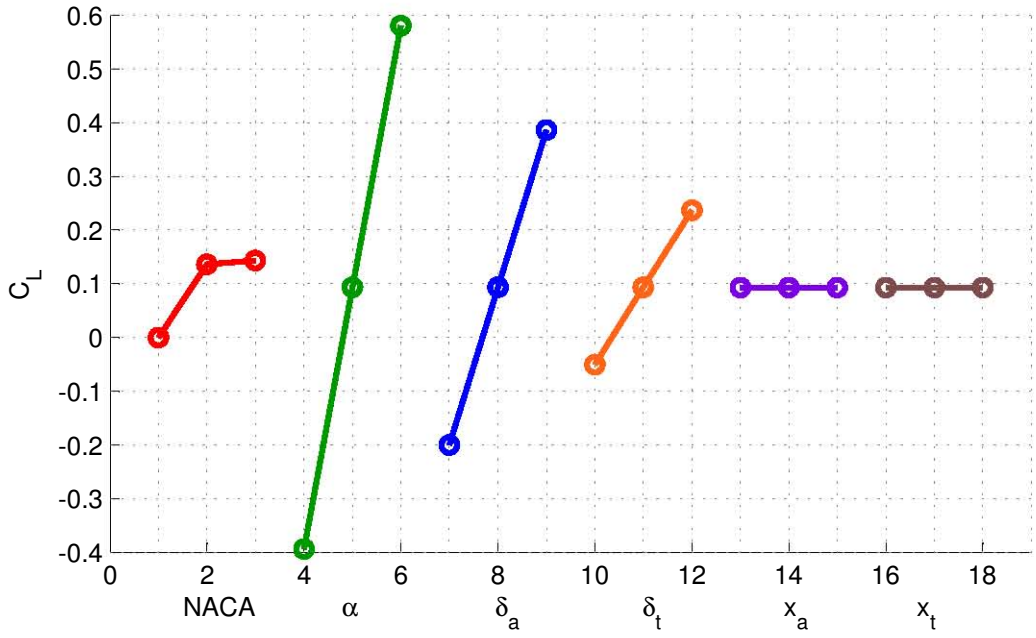


Figure 10.1: Factor effects for C_L. Full-factorial analysis.

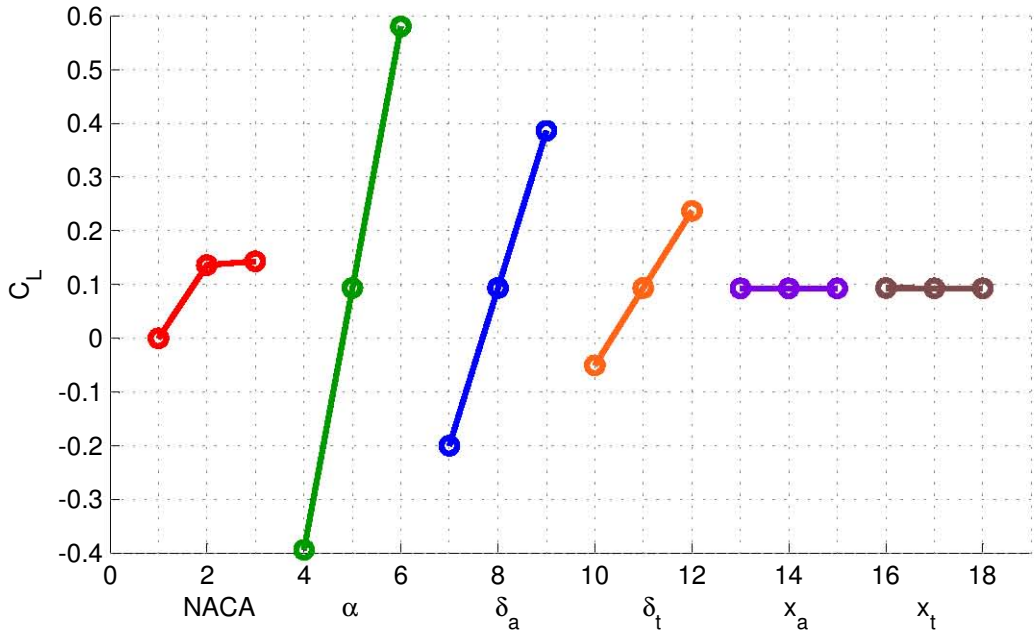


Figure 10.2: Factor effects for C_L. Fractional-factorial analysis.

Table 10.4: Percentage difference in average values for factor effects – C_L

	NACA	α	δ_a	δ_t	x_a	x_t
High	0.006%	0.0004%	0.004%	0.022%	0.277%	101.987%
Mid	0.043%	0.003%	0.001%	0.001%	0.097%	0.004%
Low	0.003%	0.001%	0.001%	0.022%	0.179%	0.004%

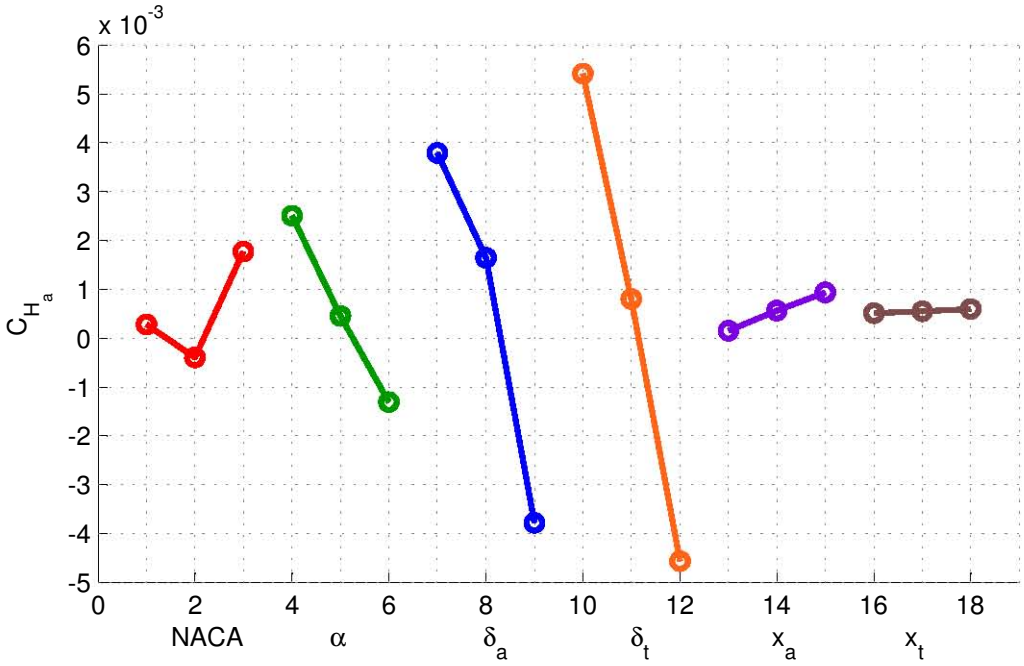


Figure 10.3: Factor effects for C_{H_a}. Full-factorial analysis.

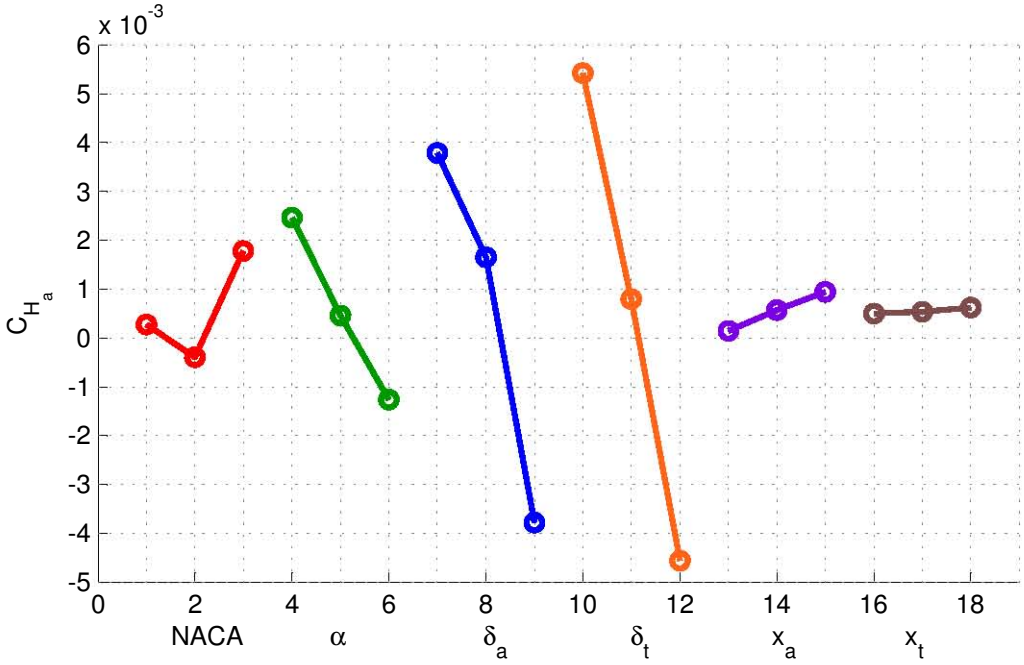


Figure 10.4: Factor effects for C_{H_a}. Fractional-factorial analysis.

Table 10.5: Percentage difference in average values for factor effects – C_{H_a}

	NACA	α	δ_a	δ_t	x_a	x_t
High	1.859%	0.181%	0.111%	1.109%	2.062%	1.345%
Mid	1.572%	0.217%	0.797%	0.623%	2.149%	0.096%
Low	3.222%	0.155%	0.065%	0.076%	4.181%	0.379%

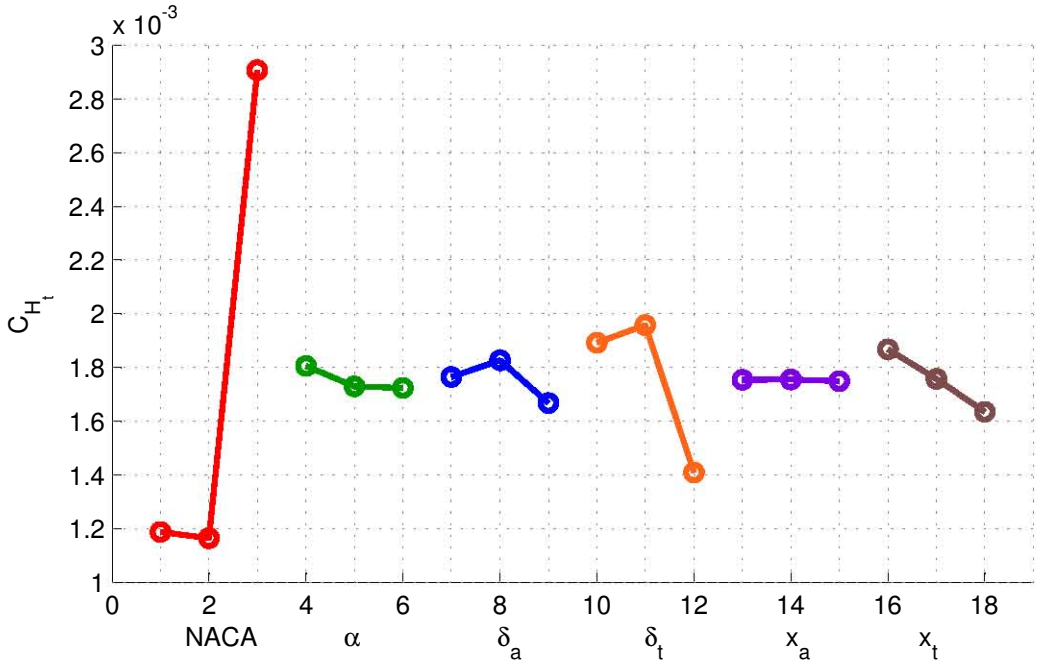


Figure 10.5: Factor effects for C_{H_t}. Full-factorial analysis.

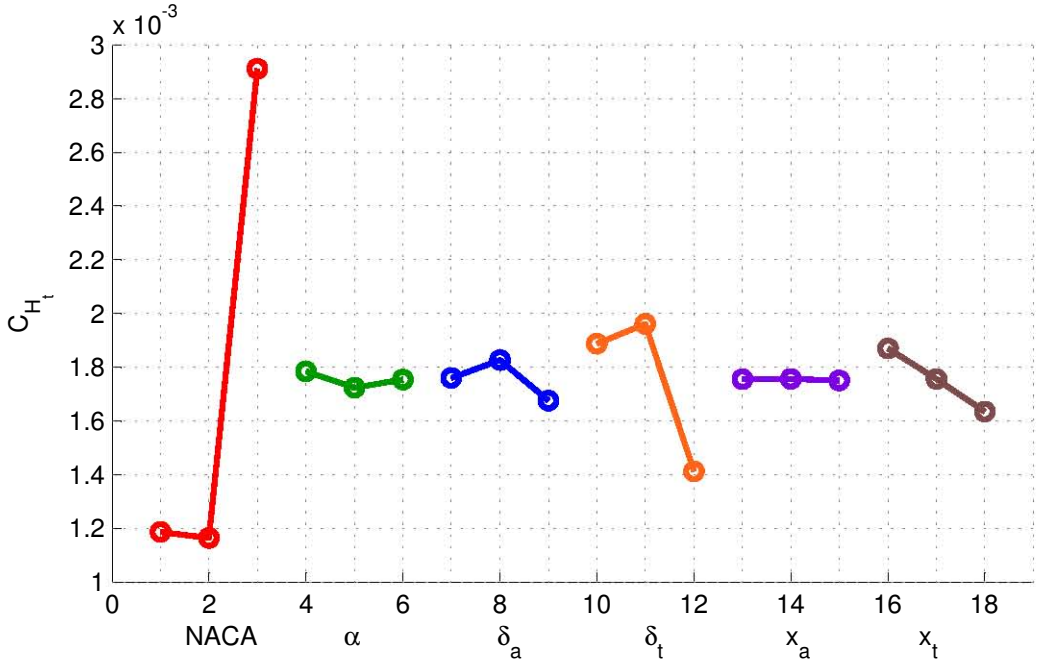


Figure 10.6: Factor effects for C_{H_t}. Fractional-factorial analysis.

Table 10.6: Percentage difference in average values for factor effects – C_{H_t}

	NACA	α	δ_a	δ_t	x_a	x_t
High	1.180%	0.287%	0.214%	0.078%	0.164%	0.066%
Mid	0.322%	0.029%	0.195%	0.081%	0.018%	0.017%
Low	1.756%	0.476%	0.258%	0.035%	0.040%	0.151%

Table 10.7: Average values for factor effects calculations – C_L

		NACA	α	δ_a	δ_t	x_a	x_t
Full	High	-3.94×10^{-01}	-2.00×10^{-01}	-5.04×10^{-02}	9.32×10^{-02}	9.32×10^{-02}	2.67×10^{-08}
	Mid	9.33×10^{-02}	9.35×10^{-02}	9.34×10^{-02}	9.32×10^{-02}	9.32×10^{-02}	1.36×10^{-01}
	Low	5.80×10^{-01}	3.86×10^{-01}	2.37×10^{-01}	9.32×10^{-02}	9.32×10^{-02}	1.43×10^{-01}
L_{81}	High	-3.94×10^{-01}	-2.00×10^{-01}	-5.04×10^{-02}	9.32×10^{-02}	9.35×10^{-02}	5.40×10^{-08}
	Mid	9.33×10^{-02}	9.35×10^{-02}	9.34×10^{-02}	9.32×10^{-02}	9.31×10^{-02}	1.36×10^{-01}
	Low	5.80×10^{-01}	3.86×10^{-01}	2.37×10^{-01}	9.32×10^{-02}	9.31×10^{-02}	1.43×10^{-01}

Table 10.8: Average values for factor effects calculations – C_{H_a}

		NACA	α	δ_a	δ_t	x_a	x_t
Full	High	2.51×10^{-03}	3.79×10^{-03}	5.42×10^{-03}	1.50×10^{-04}	5.10×10^{-04}	2.80×10^{-04}
	Mid	4.54×10^{-04}	1.65×10^{-03}	8.05×10^{-04}	5.67×10^{-04}	5.51×10^{-04}	-3.97×10^{-04}
	Low	-1.31×10^{-03}	-3.78×10^{-03}	-4.56×10^{-03}	9.40×10^{-04}	5.97×10^{-04}	1.77×10^{-03}
L_{81}	High	2.46×10^{-03}	3.79×10^{-03}	5.42×10^{-03}	1.49×10^{-04}	4.99×10^{-04}	2.76×10^{-04}
	Mid	4.61×10^{-04}	1.65×10^{-03}	7.99×10^{-04}	5.71×10^{-04}	5.40×10^{-04}	-3.97×10^{-04}
	Low	-1.26×10^{-03}	-3.78×10^{-03}	-4.56×10^{-03}	9.41×10^{-04}	6.21×10^{-04}	1.78×10^{-03}

Table 10.9: Average values for factor effects calculations – C_{H_t}

		NACA	α	δ_a	δ_t	x_a	x_t
Full	High	1.81×10^{-03}	1.76×10^{-03}	1.89×10^{-03}	1.75×10^{-03}	1.87×10^{-03}	1.19×10^{-03}
	Mid	1.73×10^{-03}	1.83×10^{-03}	1.96×10^{-03}	1.75×10^{-03}	1.76×10^{-03}	1.16×10^{-03}
	Low	1.72×10^{-03}	1.67×10^{-03}	1.41×10^{-03}	1.75×10^{-03}	1.63×10^{-03}	2.91×10^{-03}
L_{81}	High	1.78×10^{-03}	1.76×10^{-03}	1.89×10^{-03}	1.76×10^{-03}	1.87×10^{-03}	1.19×10^{-03}
	Mid	1.72×10^{-03}	1.83×10^{-03}	1.96×10^{-03}	1.76×10^{-03}	1.76×10^{-03}	1.16×10^{-03}
	Low	1.75×10^{-03}	1.68×10^{-03}	1.41×10^{-03}	1.75×10^{-03}	1.64×10^{-03}	2.91×10^{-03}

10.3 Interactions

This section contains plots of the interactions from the full-factorial analysis of C_L. The code used to generate the plots appears in Appendix J.

The coefficients of each interaction appear in parentheses in each caption in terms of the coded variables to show the significance of the term. Here coefficients from a full quadratic model were used to include all interactions, merely for the purpose of this example. Two coefficients appear for the interactions involving the term NACA, as can be seen in Table 10.1.

It can be seen that the interactions that are graphically visible have the highest coefficients. Figures 10.11, 10.13, 10.15, 10.17 and 10.18 show some interaction effects and have coefficients with absolute values ranging from 1.23×10^{-2} to 1.65×10^{-3}. All other plots have coefficients with absolute values ranging from 2.72×10^{-6} to 2.22×10^{-5}.

Although Fig. 10.19 shows an interaction it has a low coefficient. Examination of the values of C_L compared with all other plots show that some scaling has taken place, which exaggerates the appearance of the interaction. The three lines initially appeared to exactly coincide at typical scaling levels.

We can also examine the final equation to see which interaction terms were picked up in the ANOVA analysis. The equation for the 00012[1] airfoil is repeated below for convenience. It can be seen that the $\delta_a\, x_a$ interaction picked up graphically has the highest coefficient by several orders of magnitude. All other interactions seen in the plots involve NACA and so do not appear in the equations. However, several other terms were included where it is not graphically evident that an interaction is present. Therefore the graphical analysis was confined just to the C_L modelling using the full-factorial analysis, as the ANOVA results much more accurately convey this information.

$$\begin{aligned} C_{L_{00012}} = &-1.88 \times 10^{-03} + 1.19 \times 10^{-01}\alpha + 1.81 \times 10^{-01}\delta_a + 3.38 \times 10^{-01}\delta_t \\ &+ 1.11 \times 10^{-03}x_a + 1.72 \times 10^{-03}x_t - 7.68 \times 10^{-06}\alpha\,\delta_a - 5.67 \times 10^{-06}\alpha\,\delta_t \\ &+ 1.06 \times 10^{-03}\alpha\,x_a - 1.44 \times 10^{-01}\delta_a\,x_a - 3.21 \times 10^{-01}\delta_t\,x_t \\ &- 1.05 \times 10^{-05}\alpha^2 - 2.88 \times 10^{-05}\delta_a^2 - 1.55 \times 10^{-05}\delta_t^2 \end{aligned} \tag{9.3}$$

[1] The other airfoil shapes had the same coefficients for all interacting terms.

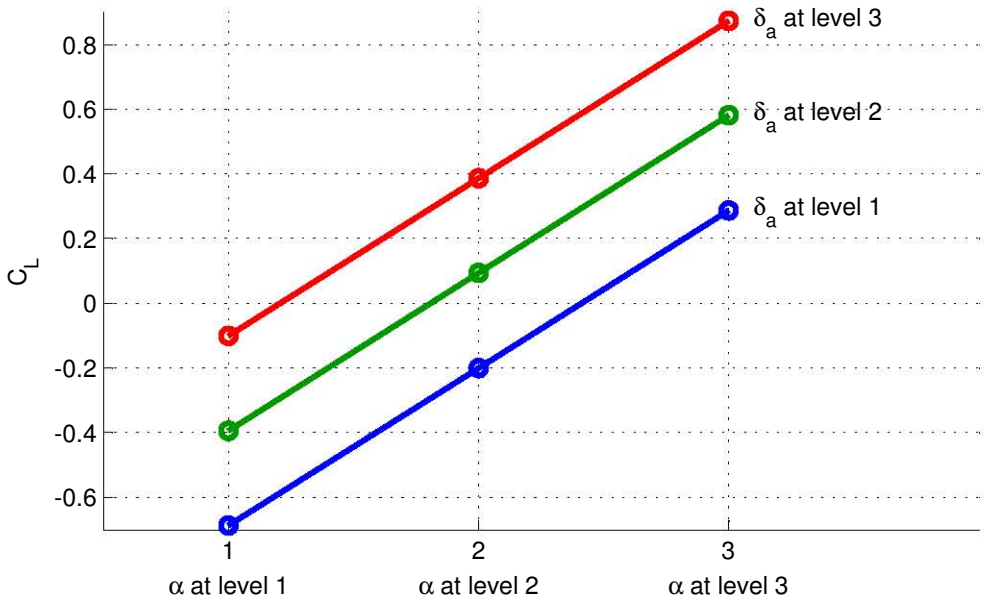


Figure 10.7: Interaction between α and δ_a. (-1.23×10^{-4})

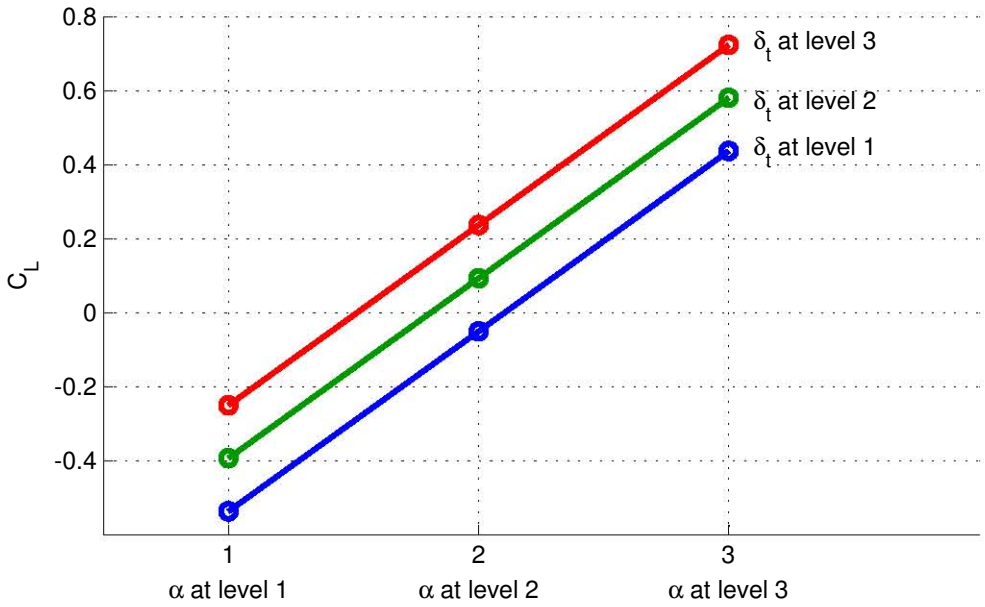


Figure 10.8: Interaction between α and δ_t. (-9.07×10^{-5})

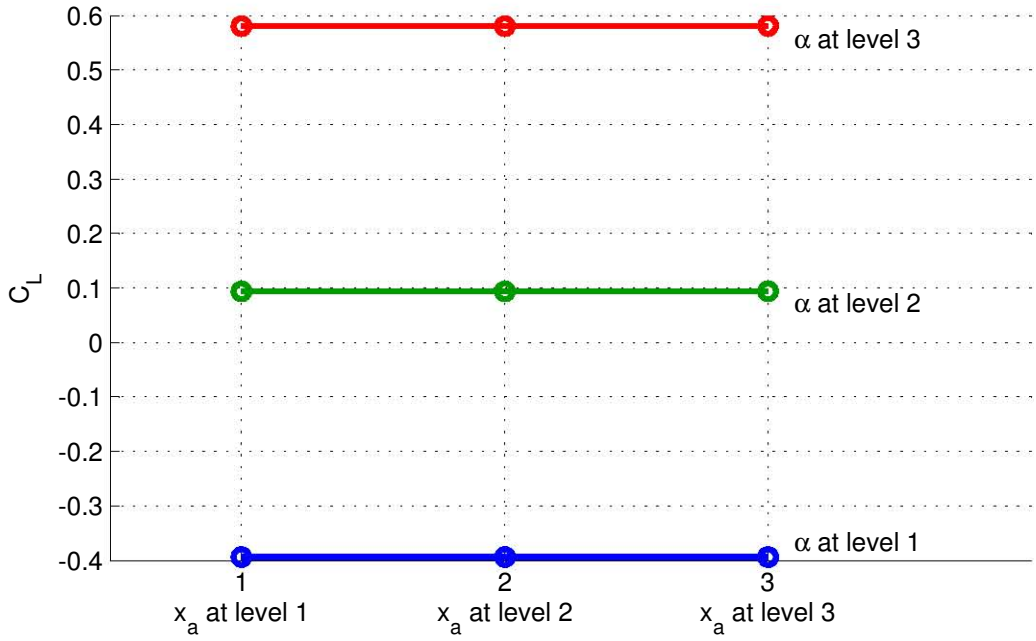


Figure 10.9: Interaction between α and x_a. (8.49×10^{-5})

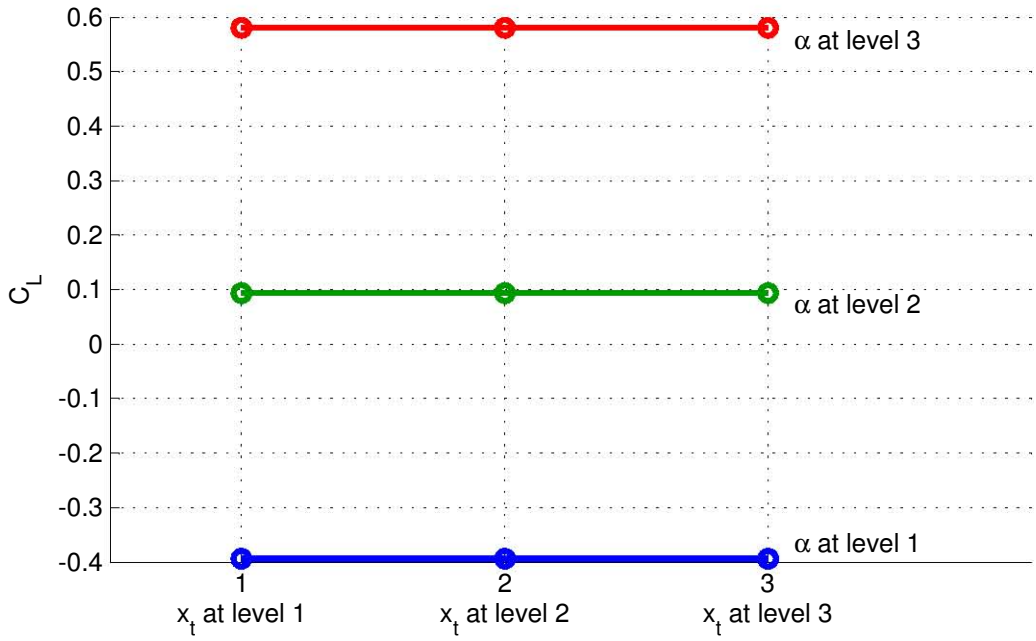


Figure 10.10: Interaction between α and x_t. (4.18×10^{-5})

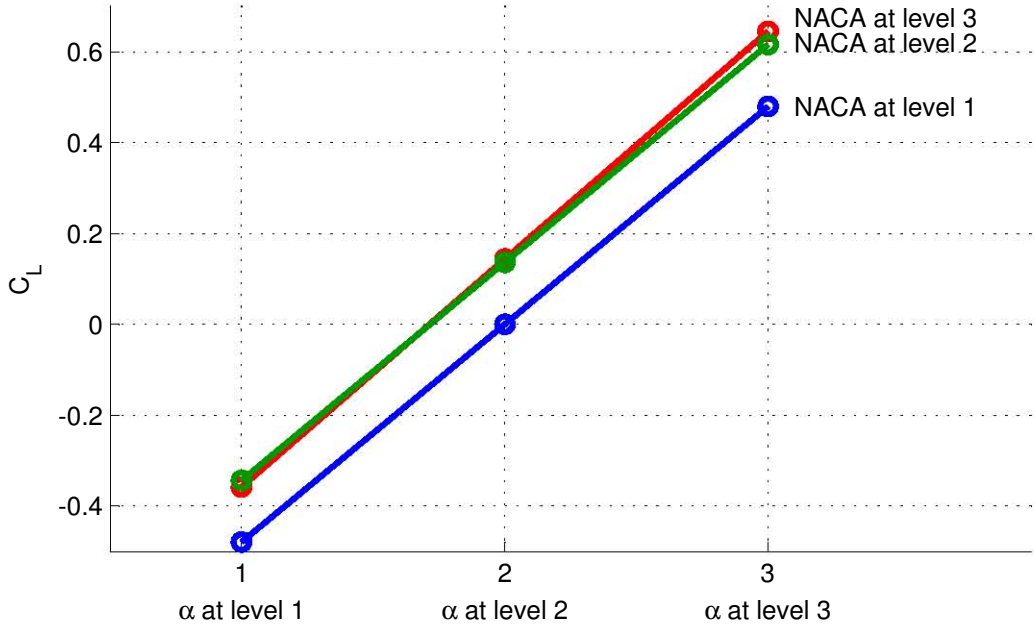


Figure 10.11: Interaction between α and NACA. $(-7.35 \times 10^{-3}, -7.30 \times 10^{-3})$

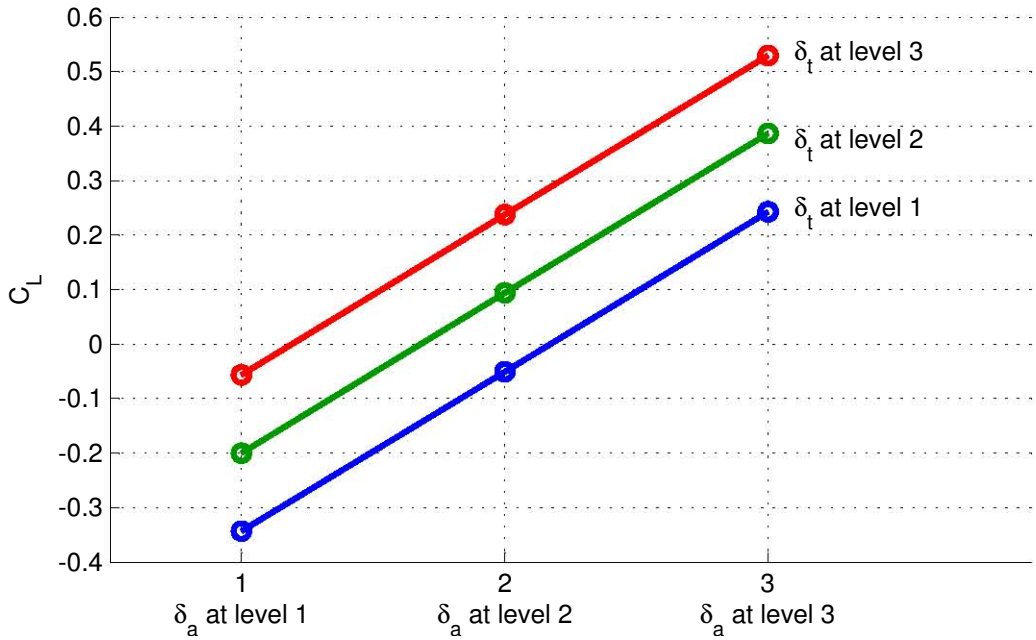


Figure 10.12: Interaction between δ_a and δ_t. (-8.76×10^{-6})

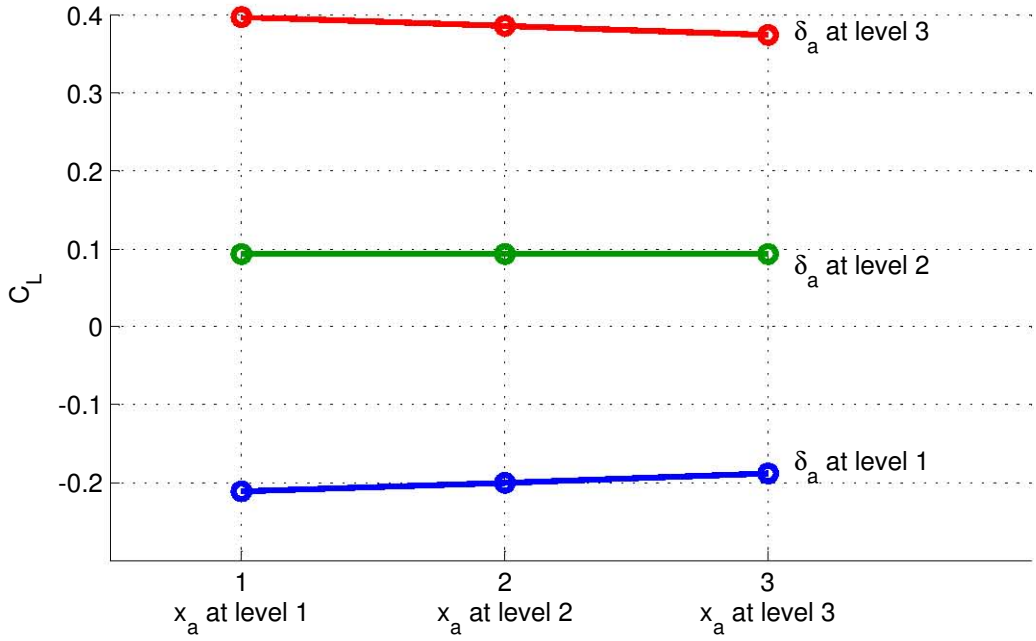


Figure 10.13: Interaction between δ_a and x_a. (-1.15×10^{-2})

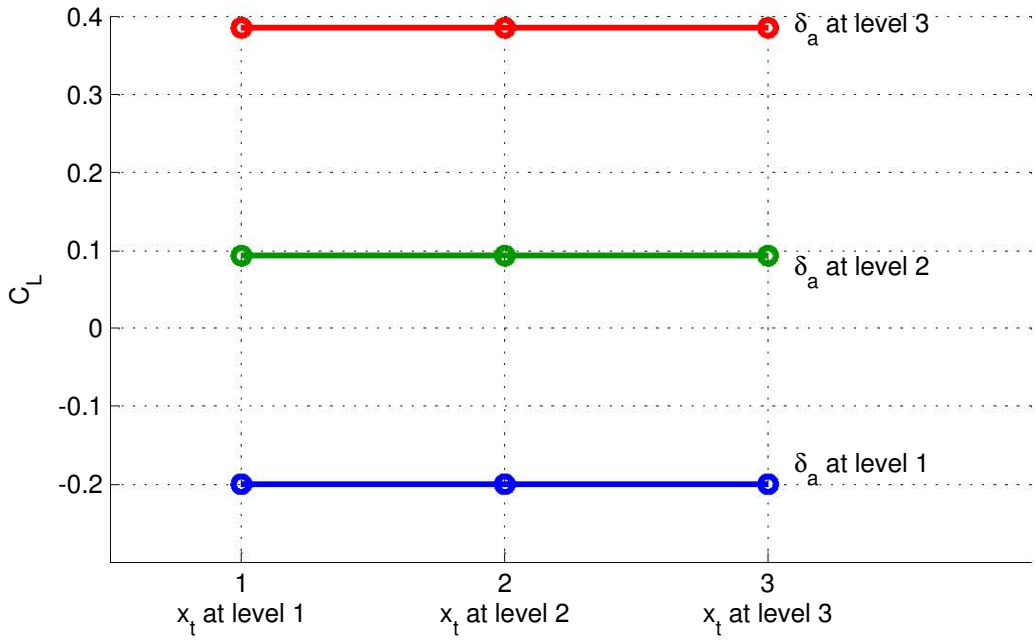


Figure 10.14: Interaction between δ_a and x_t. (2.31×10^{-5})

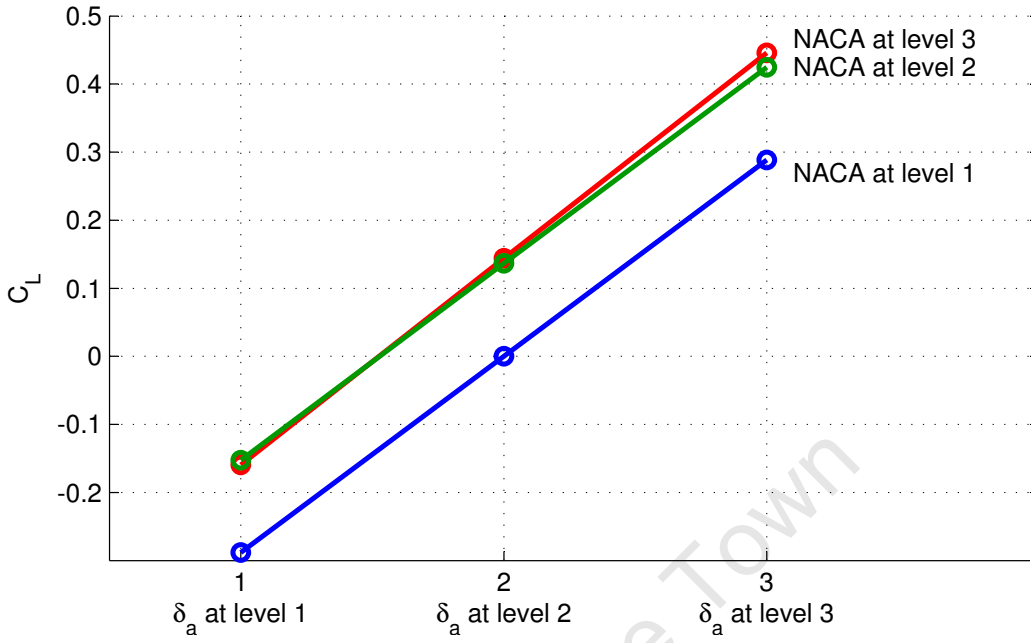


Figure 10.15: Interaction between δ_a and NACA. $(-4.76 \times 10^{-3}, -4.71 \times 10^{-3})$

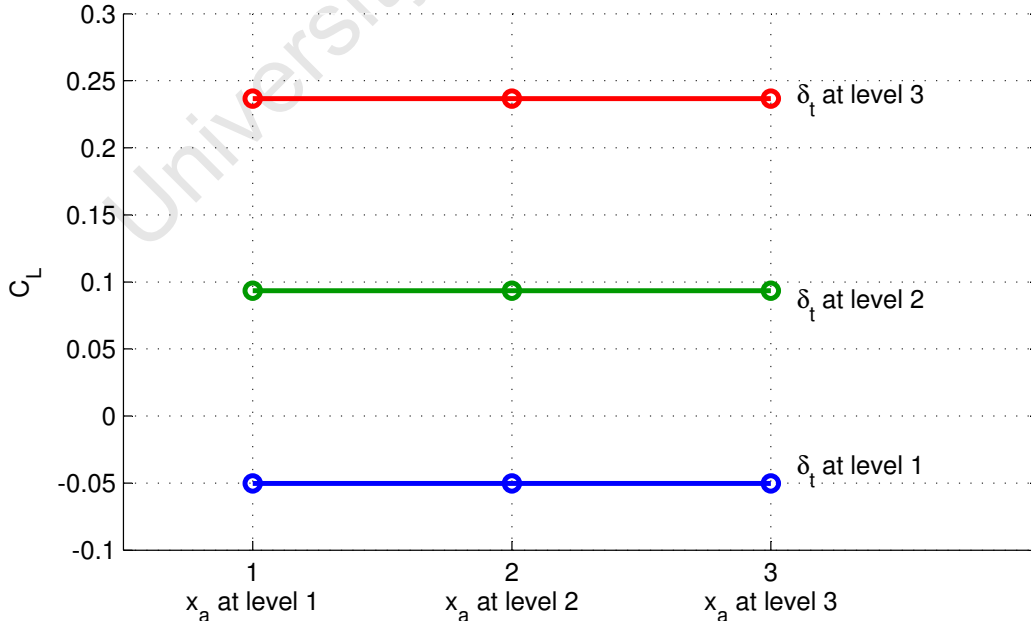


Figure 10.16: Interaction between δ_t and x_a. (1.22×10^{-5})

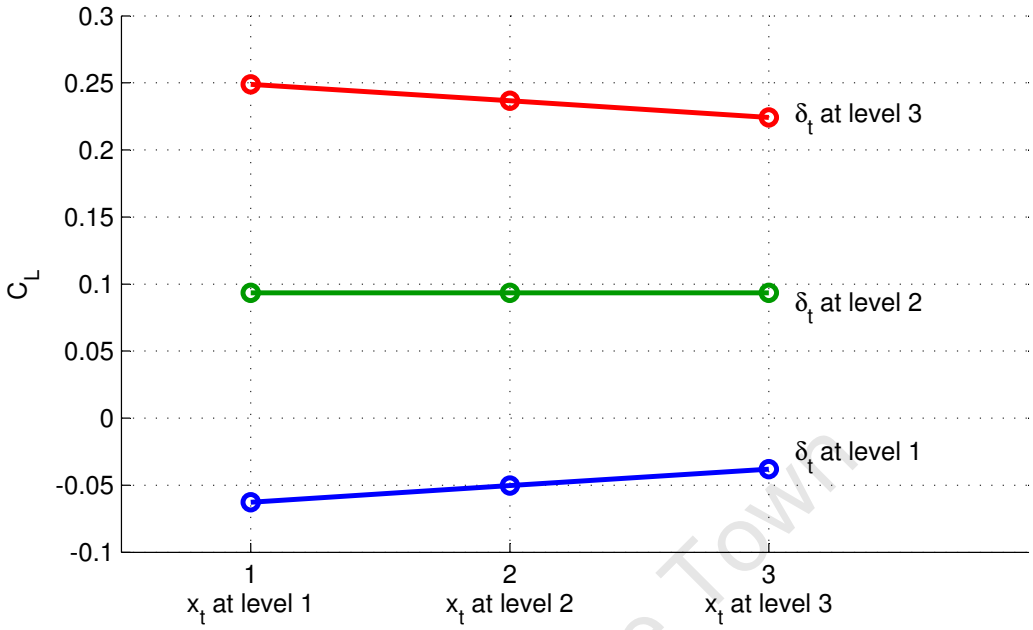


Figure 10.17: Interaction between δ_t and x_t. (-1.23×10^{-2})

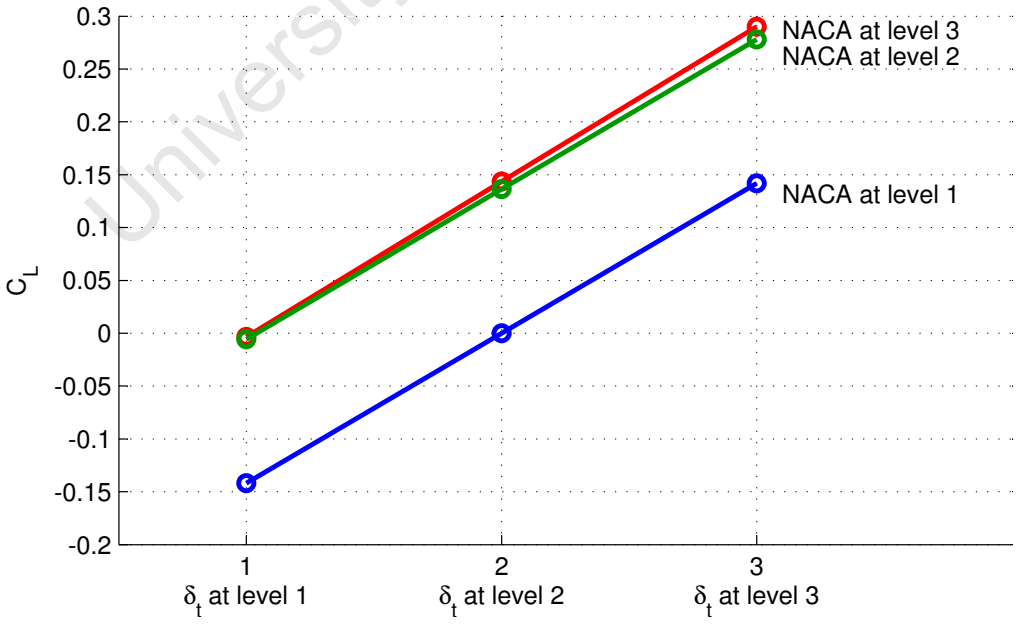


Figure 10.18: Interaction between δ_t and NACA. $(-1.65 \times 10^{-3}, -1.63 \times 10^{-3})$

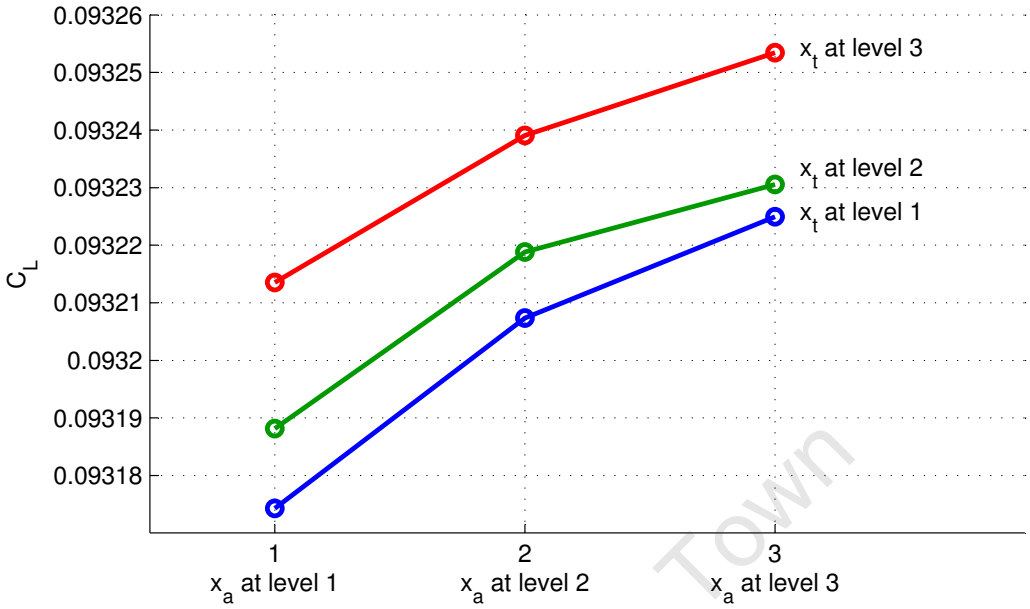


Figure 10.19: Interaction between x_a and x_t. (-2.72×10^{-6})

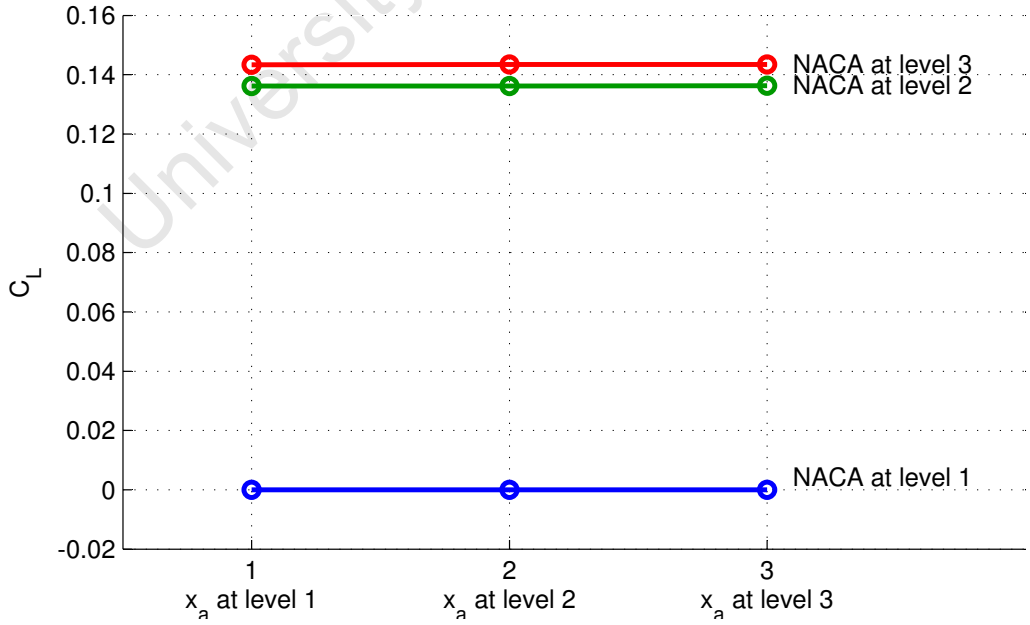


Figure 10.20: Interaction between x_a and NACA. $(-2.22 \times 10^{-5}, 7.34 \times 10^{-6})$

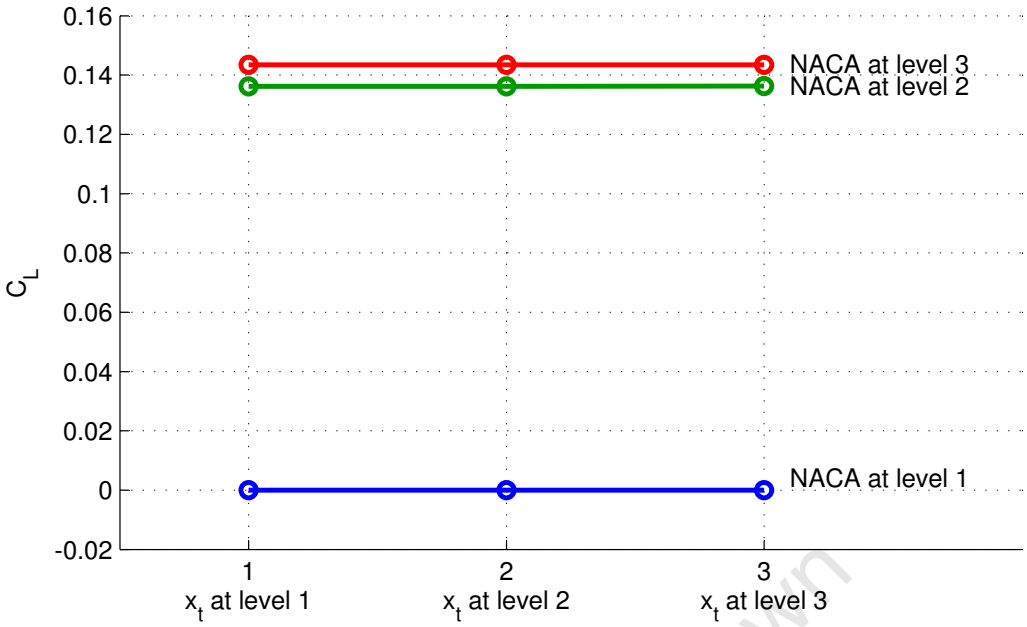


Figure 10.21: Interaction between x_t and NACA. $(-1.65 \times 10^{-5}, 7.85 \times 10^{-6})$

10.4 Merit Function Analysis

10.4.1 Calculating The Merit Function

Returning now to the mathematical models generated by Design Expert in Chapter 9, equations 9.3 to 9.20 were entered into Matlab to facilitate analysis of the merit function. The Merit function as discussed in §9.2.1 is repeated below for convenience.

$$M = \frac{C_{L_{\delta_t \neq 0}} - C_{L_{\delta_t = 0}}}{C_{H_{t,\ \delta_t \neq 0}}} \tag{9.2}$$

Appendix I contains the Matlab code used to calculate the merit function. An explanation accompanies the code, while the overall methodology and results will be described here.

A range of factor levels were chosen with minimum and maximum limits as with the previous analyses, outlined in Table 9.2. Cycling through this range of geometric configurations (of hinge locations and angle of attack) the following steps were followed in order to calculate the data for the merit function:

1. calculate δ_a for $\delta_t \neq 0$, $C_{Ha} = 0$
2. calculate δ_a for $C_{Ha} = 0$, $\delta_t = 0$
3. calculate C_L for $\delta_t = 0$
4. calculate C_L for $\delta_t \neq 0$
5. calculate C_{H_t} for $\delta_t \neq 0$

Steps 1 and 2 calculate the necessary δ_a to zero C_{Ha} (simulating the floating hinge) for the cases of an activated tab ($\delta_t \neq 0$) and a deactivated tab (where

$\delta_t = 0$). Using each δ_a, the corresponding C_L is then calculated in steps 3 and 4, and finally C_{H_t} is calculated in step 5. This yields all the necessary terms of the merit function. These are averaged over each geometric configuration to ensure that the given configuration performs best over the entire working range of the rudder.

10.4.2 Configurations With No Solution

Upon evaluation of the merit function in this manner it was found that for certain configurations no solution for δ_a was found. Figures 10.22 to 10.25 illustrate the problem. Plotting δ_a vs C_{H_a} it can be seen that certain configurations exist where the hinge moment can theoretically not be zeroed for the given tab deflection. This is because the mathematics tells us that for certain settings of α and δ_t no matter what δ_a is, C_{H_a} will always be negative. This results in the strange physical situation that the rudder will rotate in a counter-clockwise direction.

To resolve this it was found that the problem could be avoided by limiting δ_t, to a maximum of 2.797 in the fractional-factorial analysis, and 2.737 in the full-factorial analysis. However, to directly compare the merit function both analyses were limited to 2.737.

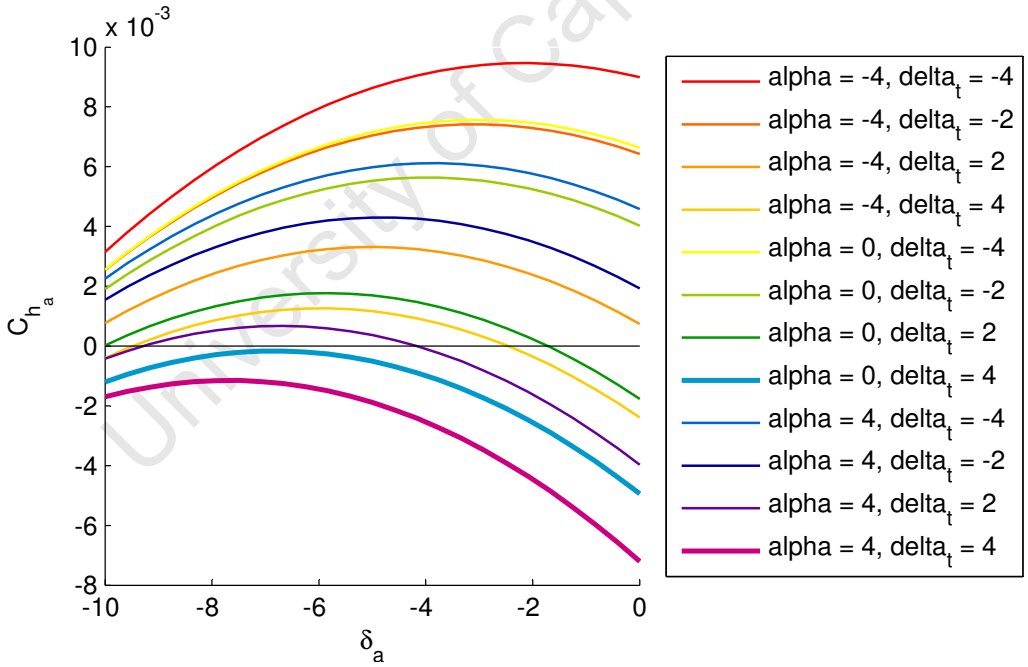


Figure 10.22: Plot of δ_a vs C_{H_a} showing two configurations where C_{H_a} cannot be zeroed. Airfoil is a NACA 00012 with $x_a = 0.74$ and $x_t = 0.9324$. The full-factorial analysis equations have been used.

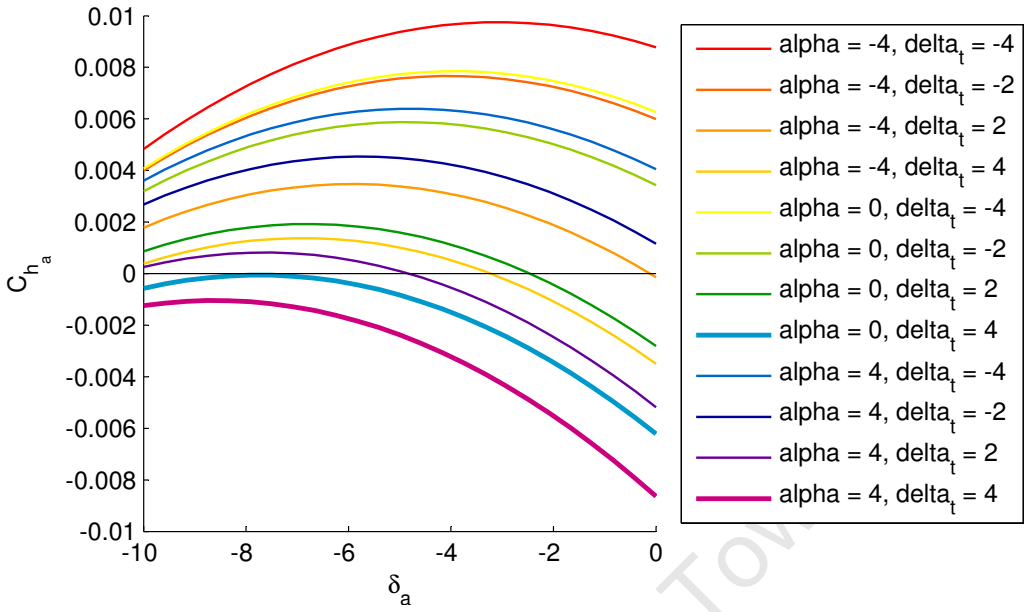


Figure 10.23: Plot of δ_a vs C_{Ha} showing a single configuration where C_{Ha} cannot be zeroed. Airfoil is a NACA 23012 with $x_a = 0.74$ and $x_t = 0.9324$. The full-factorial analysis equations have been used.

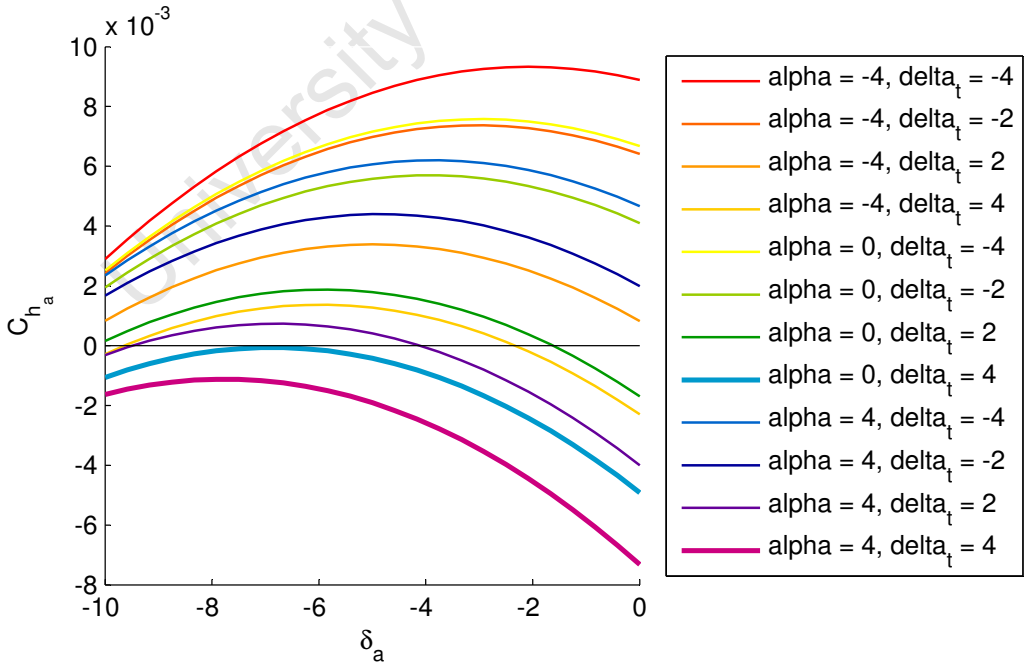


Figure 10.24: Plot of δ_a vs C_{Ha} showing two configurations where C_{Ha} cannot be zeroed. Airfoil is a NACA 00012 with $x_a = 0.74$ and $x_t = 0.9324$. The fractional-factorial analysis equations have been used.

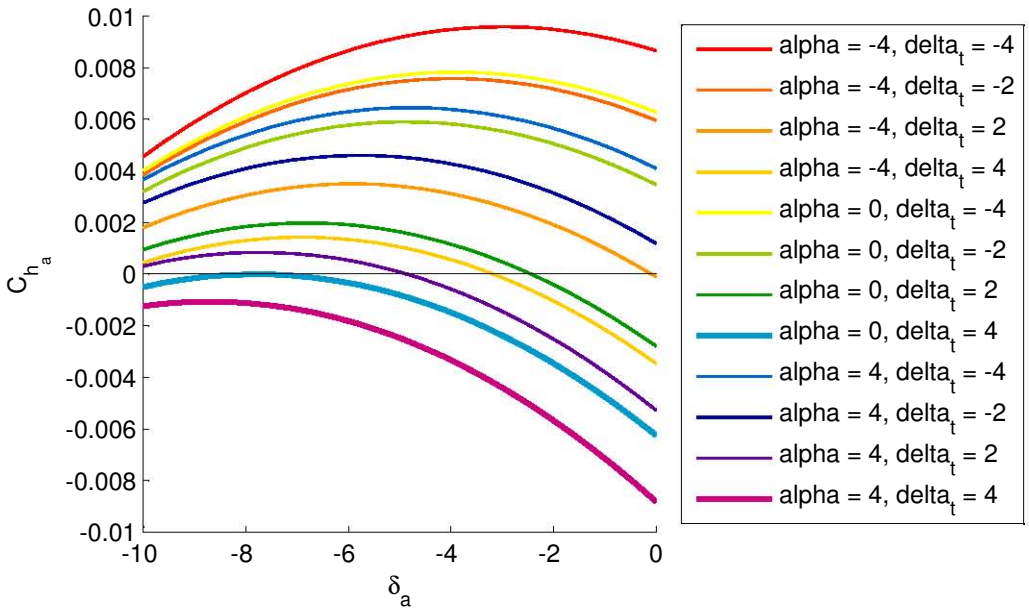


Figure 10.25: Plot of δ_a vs C_{Ha} showing a single configuration where C_{Ha} cannot be zeroed. Airfoil is a NACA 23012 with $x_a = 0.74$ and $x_t = 0.9324$. The fractional-factorial analysis equations have been used.

10.5 Merit Function Factor Effects

Plots of factor effects for the merit function are shown in Figs. 10.26 and 10.27 for the full- and fractional-factorial analyses respectively. These were generated by the code in Appendix I. It can be seen that results are very similar. It can be inferred from these plots that a thin, cambered airfoil performs best, while x_a should be moved backward and x_t forward, with the best performance found at high negative angles of attack.

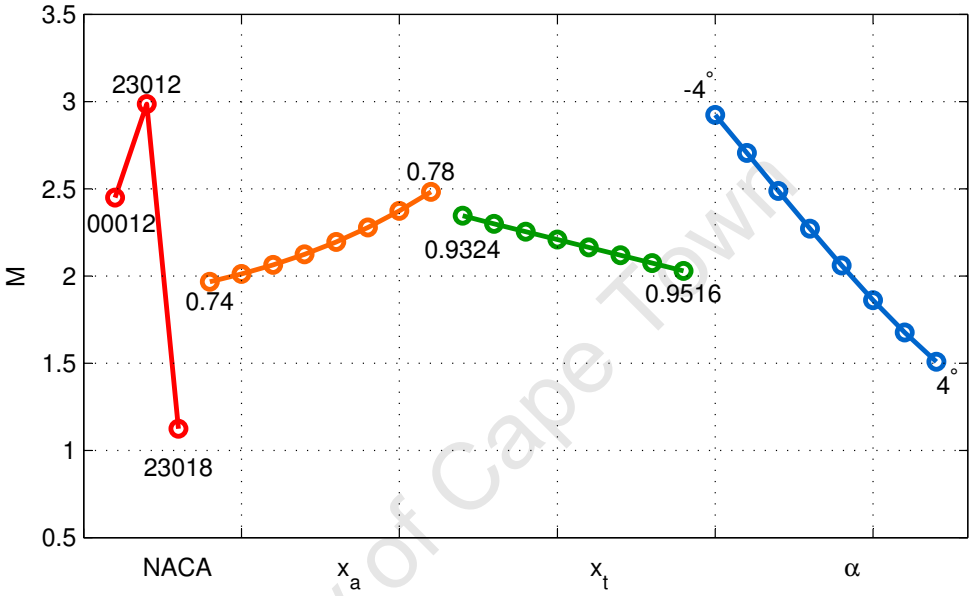


Figure 10.26: Merit function factor effects for full-factorial analysis

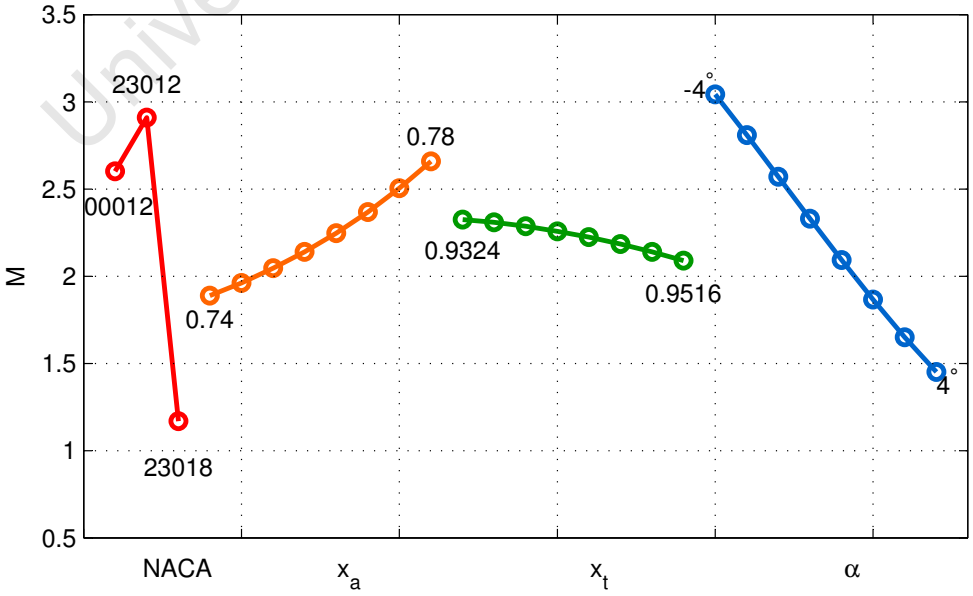


Figure 10.27: Merit function factor effects for fractional-factorial analysis

10.6 L_{81} Accuracy Analysis

In order to examine the relative accuracy of the L_{81} equations a graphical analysis of the residuals of each response was performed, for both full- and fractional-factorial equations, where a direct comparison of the results can be made. §10.6.1 contains the L_{81} residual analysis including a brief discussion of the results, while §10.6.2 contains the full-factorial residual analysis without discussion, merely to facilitate comparison.

10.6.1 L_{81} Residual analysis

C_L

Figure 10.28 shows a scatter plot of d_i vs. C_L. The distribution of data shows a slight pattern indicating a slightly poor model. Several outliers can be seen falling beyond the interval $-3 < d_i < 3$. Table 10.10 shows the treatment combinations and the recorded and predicted responses for these instances. It can be seen that although the airfoil and position of the tab hinge are varied, these outliers occurred at the extremes of lift, where α, δ_a and δ_t were either all positive or all negative, although the aileron hinge was at its smallest. This is unsurprising, as it can be expected that the mathematical model would degrade at the extremes.

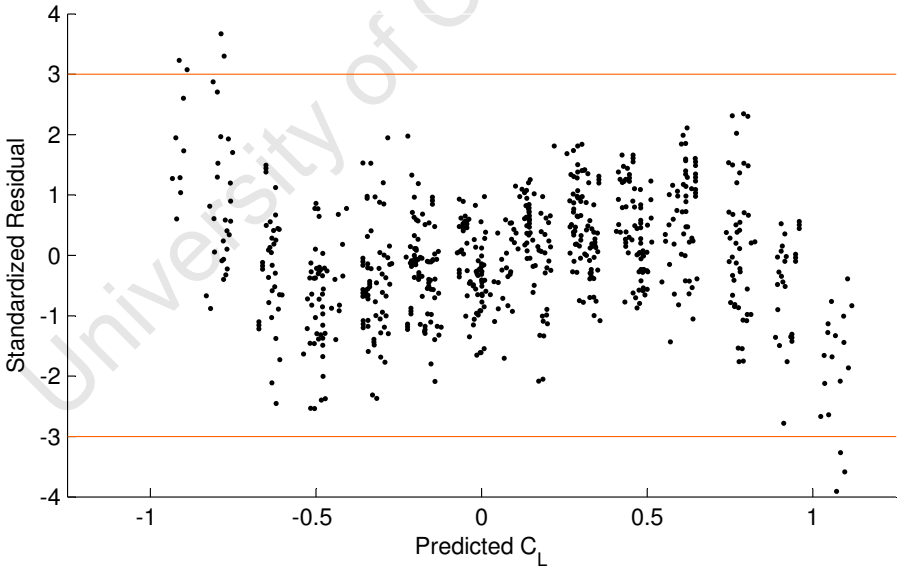


Figure 10.28: Scatter plot of standardised residual, d_i, vs. the predicted value of C_L

Figures 10.29 and 10.30 show the normal % probability distribution of the standardised residual and the histogram of the residual respectively. Both show fairly good agreement with the expected appearances of these graphs, and therefore indicate a fairly accurate model.

Table 10.10: Outliers of C_L analysis

Run	α	δ_a	δ_t	$x_{h,\,a}$	$x_{h,\,t}$	NACA	y_i	$\widehat{y_i}$

19	-4	-4	-4	0.78	0.9324	10012	-9.09E-01	-9.13E-01
20	-4	-4	-4	0.78	0.9324	23012	-7.74E-01	-7.77E-01
25	-4	-4	-4	0.78	0.9516	10012	-8.85E-01	-8.88E-01
27	-4	-4	-4	0.78	0.9516	23018	-7.82E-01	-7.86E-01
723	4	4	4	0.78	0.9324	23018	1.093E+00	1.096E+00
726	4	4	4	0.78	0.9424	23018	1.080E+00	1.084E+00
729	4	4	4	0.78	0.9516	23018	1.068E+00	1.072E+00

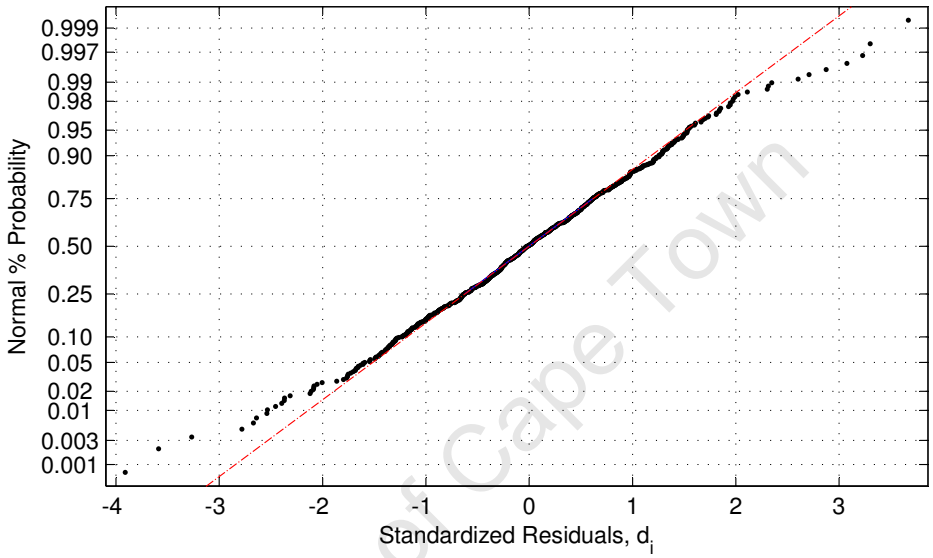


Figure 10.29: Normal % probability distribution of the standardised residual in the predicted value of C_L

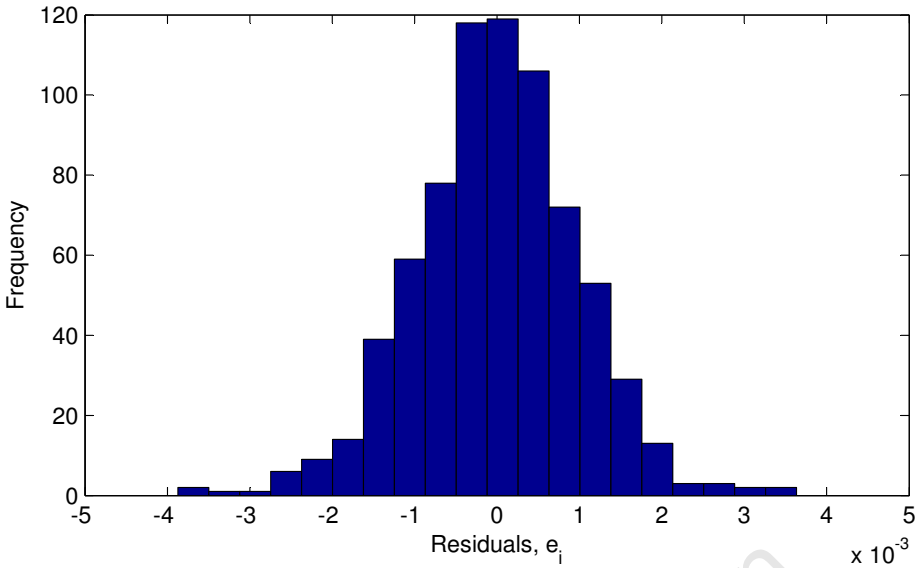


Figure 10.30: Histogram of e_i distribution for C_L

C_{Ha}

Figure 10.31 shows a scatter plot of d_i vs. C_{Ha}. The distribution of data shows a different kind of pattern than with the C_L analysis, although it still is rather slight and therefore indicates a fair model. Several outliers can be seen falling beyond the interval $-3 < d_i < 3$, with Table 10.11 showing the treatment combinations and the recorded and predicted responses for these instances. It can again be seen that these outliers occurred mostly at extreme conditions where flow over the trailing edge would be particularly strong.

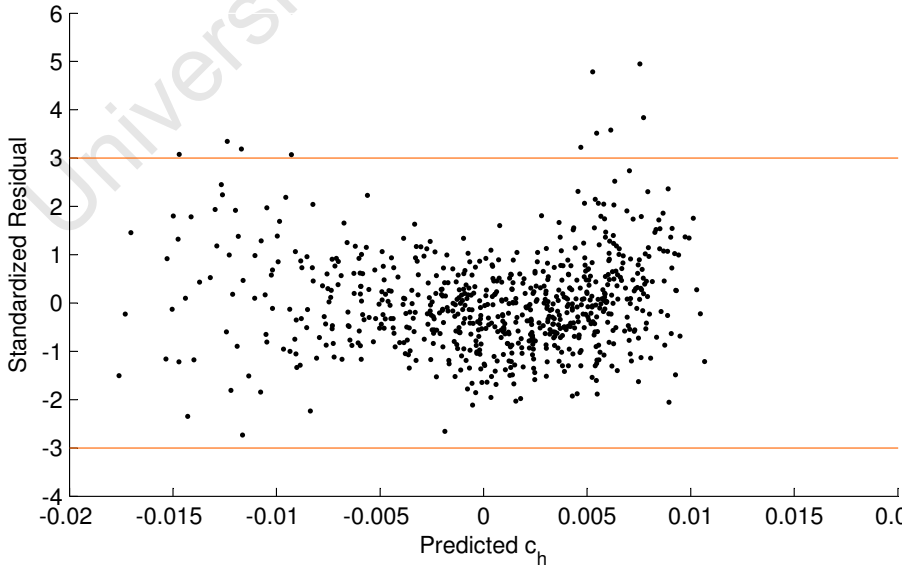


Figure 10.31: Scatter plot of Standardised Residual, d_i, vs. the predicted value of C_{Ha}

Table 10.11: Outliers of C_{Ha} analysis

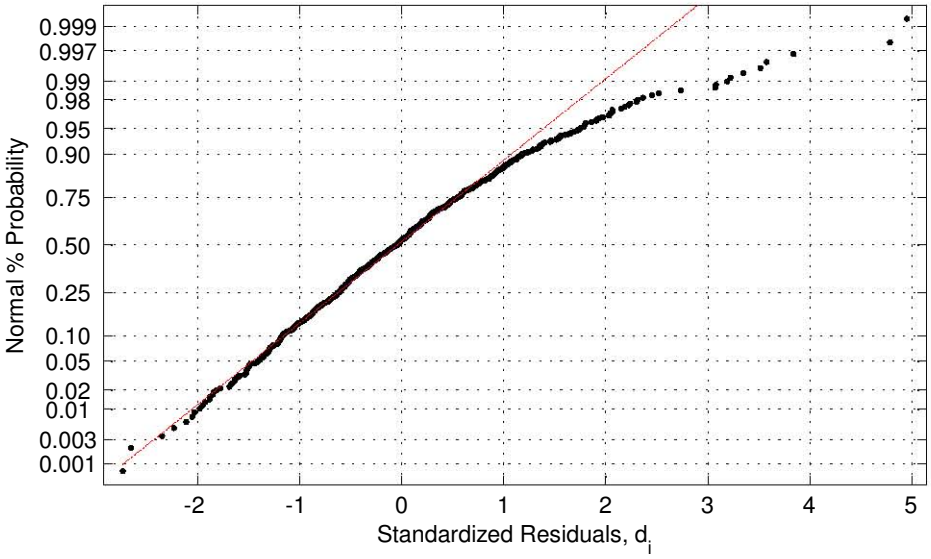


Figure 10.32: Normal % probability distribution of the standardised residual in the predicted value of C_{Ha}

Run	α	δ_a	δ_t	$x_{h,\,a}$	$x_{h,\,t}$	NACA	y_i	$\widehat{y}_i$
22	-4	-4	-4	0.78	0.9424	10012	6.04E-03	5.46E-03
24	-4	-4	-4	0.78	0.9424	23018	8.37E-03	7.73E-03
25	-4	-4	-4	0.78	0.9516	10012	6.07E-03	5.27E-03
26	-4	-4	-4	0.78	0.9516	23012	6.74E-03	6.15E-03
27	-4	-4	-4	0.78	0.9516	23018	8.37E-03	7.54E-03
268	0	-4	-4	0.78	0.9516	10012	5.23E-03	4.69E-03
719	4	4	4	0.76	0.9516	23012	-1.42E-02	-1.47E-02
720	4	4	4	0.76	0.9516	23018	-1.12E-02	-1.17E-02
728	4	4	4	0.78	0.9516	23012	-1.18E-02	-1.24E-02
729	4	4	4	0.78	0.9516	23018	-8.76E-03	-9.27E-03

Figures 10.32 and 10.33 show the normal % probability distribution of the standardised residual and the histogram of the residual respectively. Both plots show a slightly skewed distribution, with greater accuracy in one half of the data. This indicates the model is possibly inadequate.

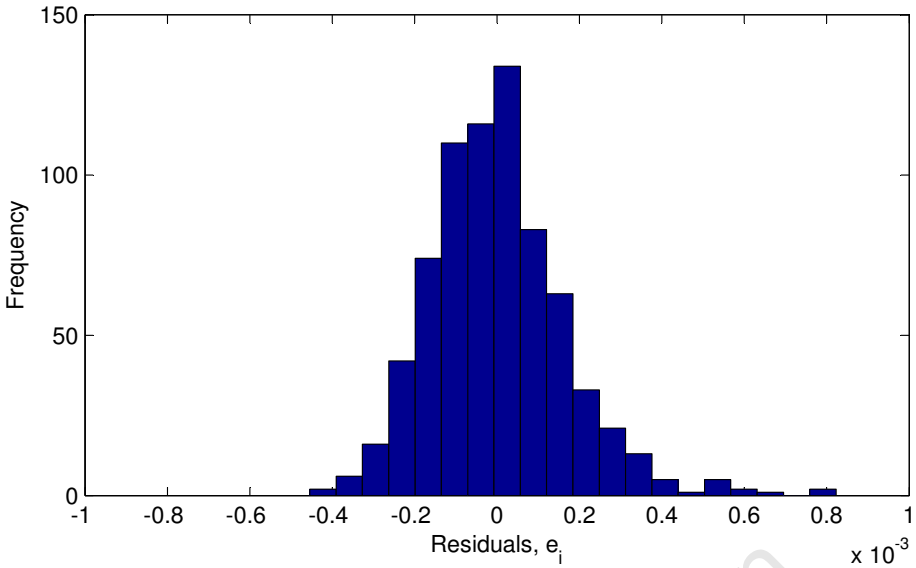


Figure 10.33: Histogram of e_i distribution for C_{H_a}

C_{H_t}

Figure 10.34 shows a scatter plot of d_i vs. C_{H_t}. The distribution of data here shows a fairly non-random nature and therefore indicates a poor model. Table 10.12 shows the treatment combinations and the recorded and predicted responses for these instances for the few outliers.

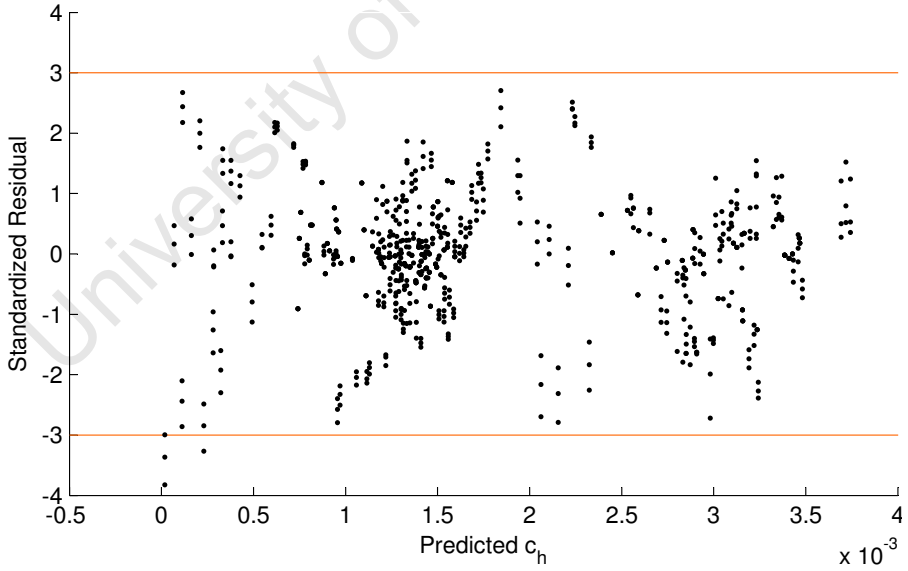


Figure 10.34: Scatter plot of Standardised Residual, d_i, vs. the predicted value of C_{H_t}

Table 10.12: Outliers of C_{H_t} analysis

Run	α	δ_a	δ_t	$x_{h,\,a}$	$x_{h,\,t}$	NACA	y_i	$\widehat{y}_i$
713	4	4	4	0.76	0.9324	23012	-2.02E-04	1.87E-05
721	4	4	4	0.78	0.9324	10012	1.57E-05	2.29E-04
722	4	4	4	0.78	0.9324	23012	-2.32E-04	1.87E-05

Figures 10.35 and 10.36 show the normal % probability distribution of the standardised residual and the histogram of the residual respectively. Although the histogram shows a fair agreement with the expected appearances, the normal % probability clearly strays from its expected appearance indicating a poor model by some degree.

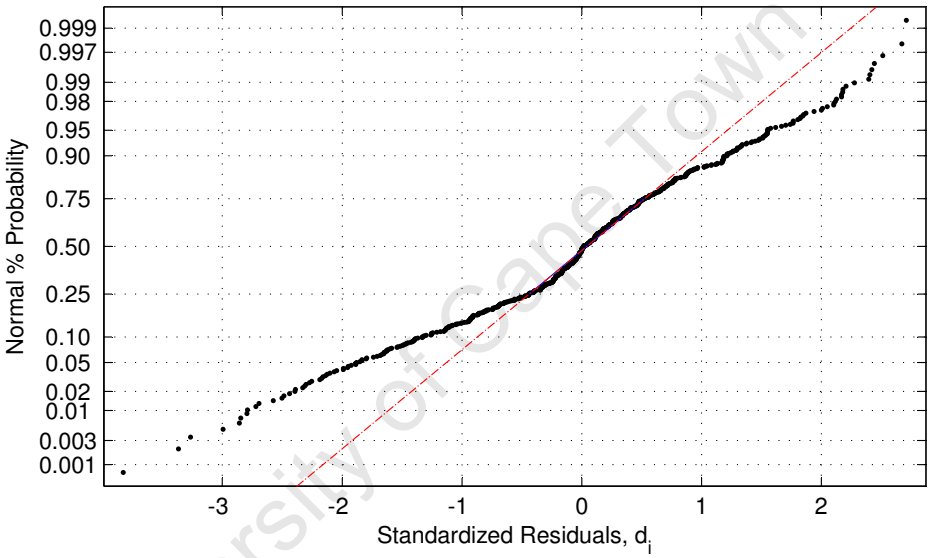


Figure 10.35: Normal % probability distribution of the standardised residual in the predicted value of C_{H_t}

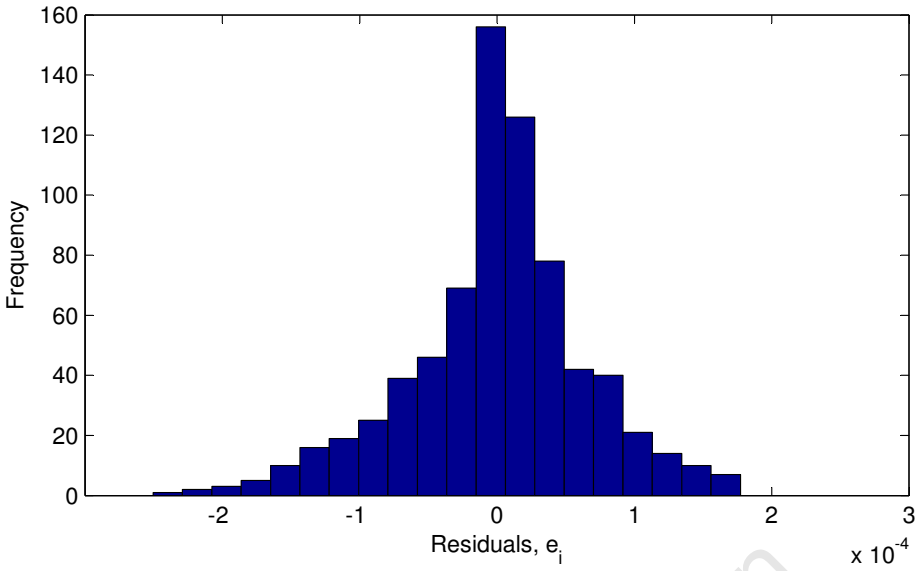


Figure 10.36: Histogram of e_i distribution for C_{H_t}

10.6.2 Full-Factorial Plots

C_L

The plots in this section were generated in a similar way to those above, but with the equations calculated during the full-factorial analysis. From the plots here the equations from the full-factorial analysis appear slightly superior.

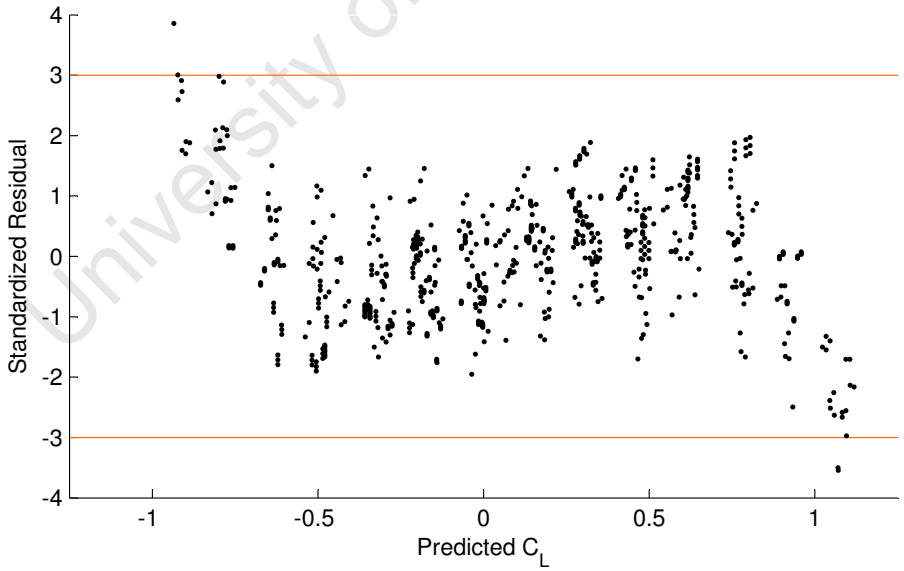


Figure 10.37: Scatter plot of standardised residual, d_i, vs. the predicted value of C_L

C_{H_a}

C_{H_t}

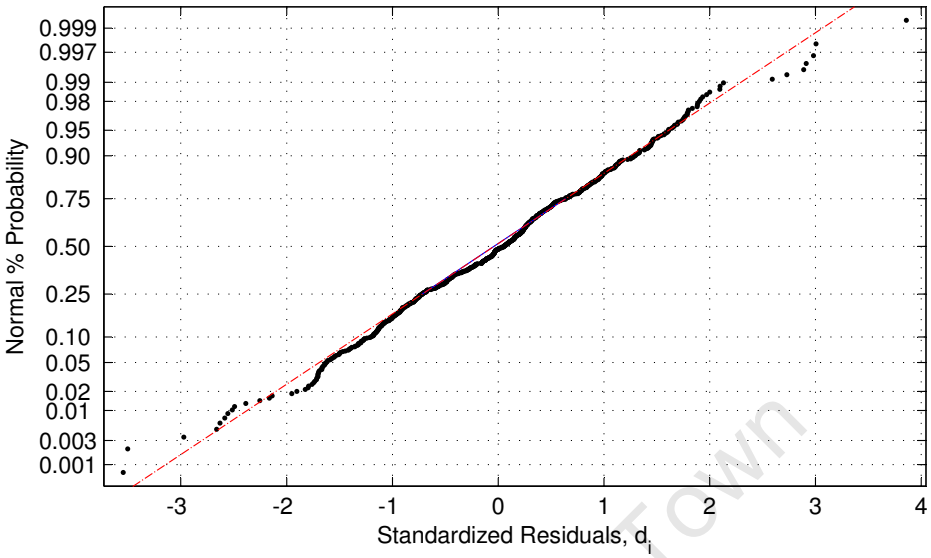


Figure 10.38: Normal % probability distribution of the standardised residual in the predicted value of C_L

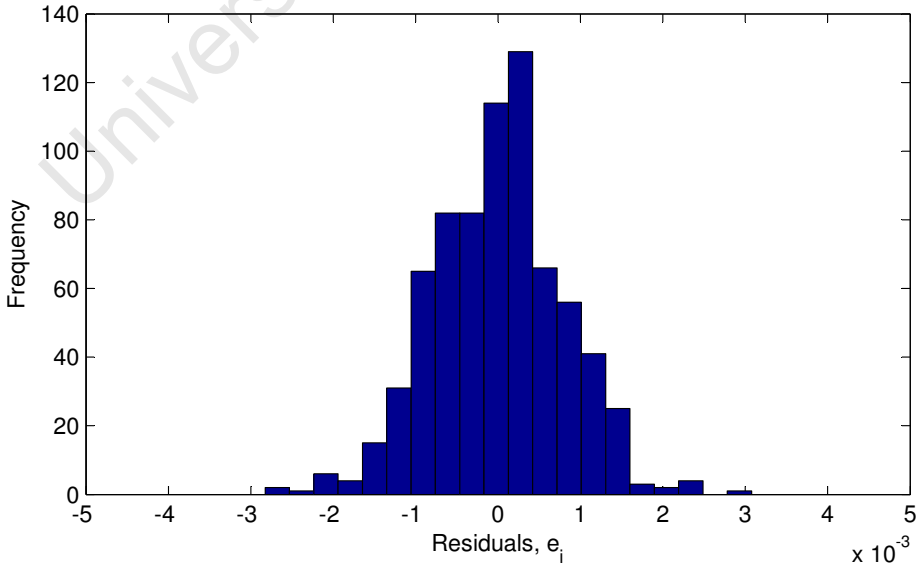


Figure 10.39: Histogram of e_i distribution for C_L

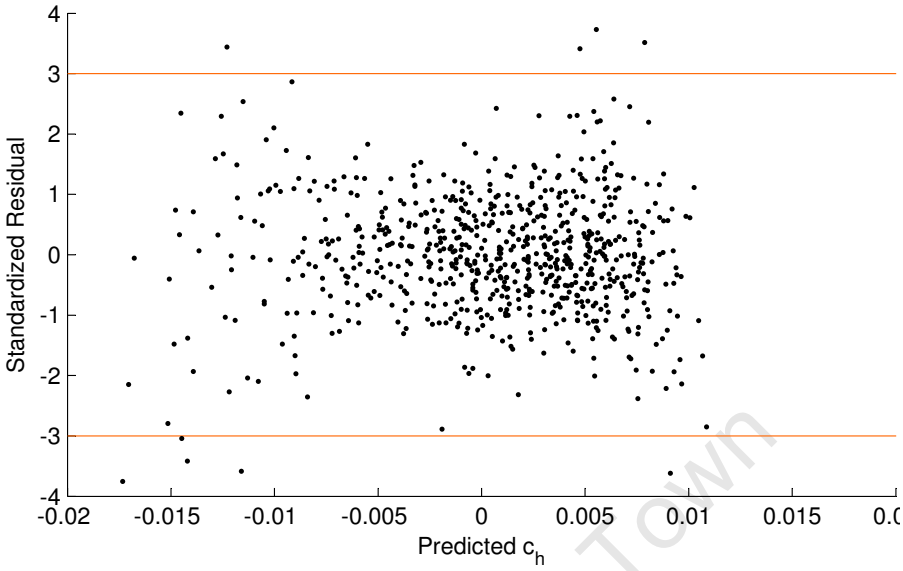


Figure 10.40: Scatter plot of Standardised Residual, d_i, vs. the predicted value of C_{Ha}

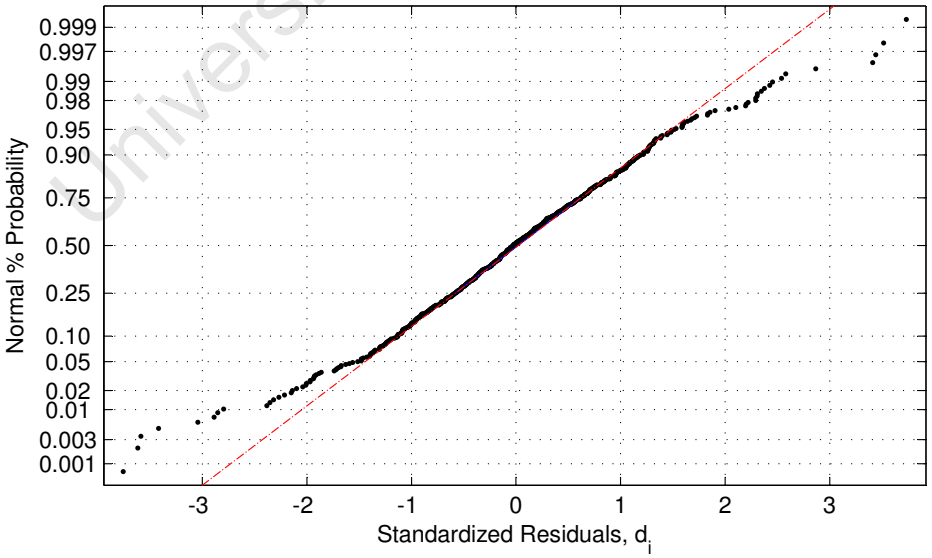


Figure 10.41: Normal % probability distribution of the standardised residual in the predicted value of C_{Ha}

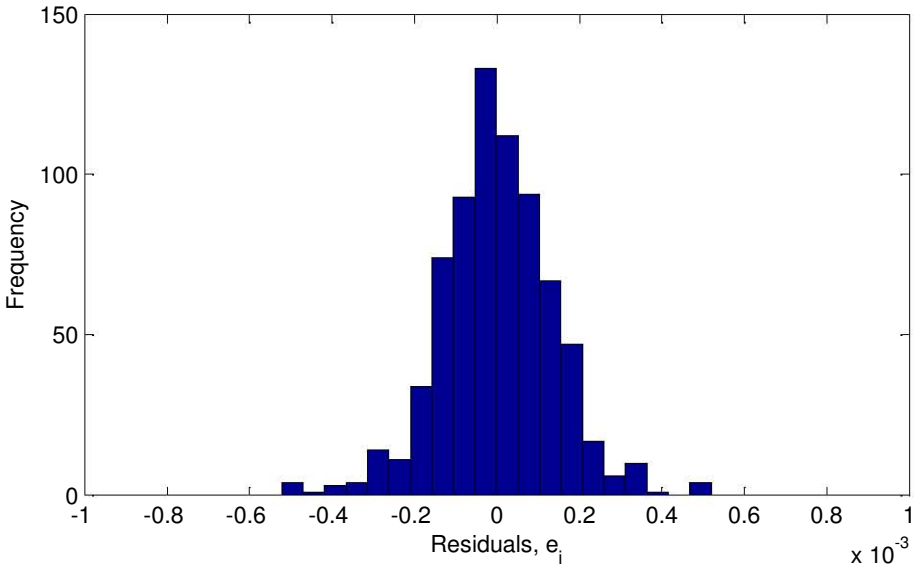


Figure 10.42: Histogram of e_i distribution for C_{H_a}

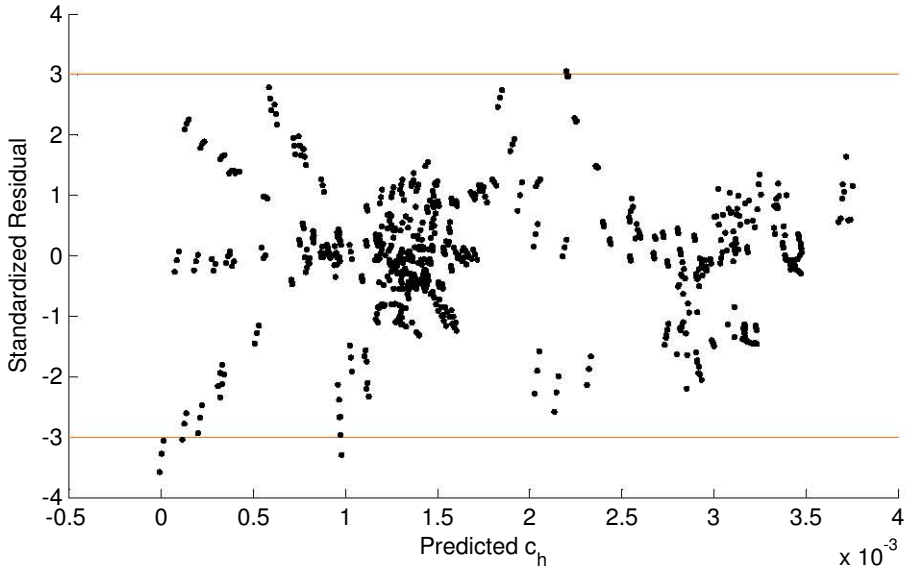


Figure 10.43: Scatter plot of Standardised Residual, d_i, vs. the predicted value of C_{H_t}

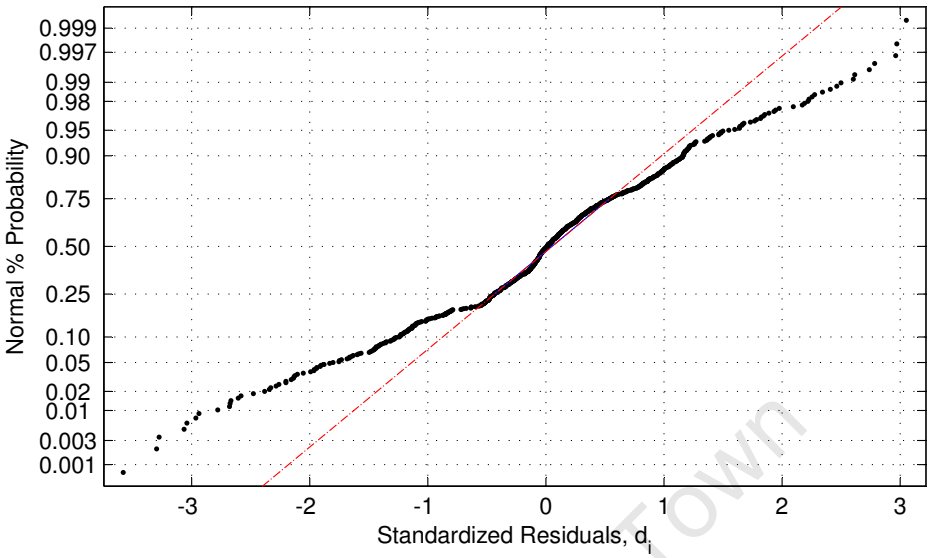


Figure 10.44: Normal % probability distribution of the standardised residual in the predicted value of C_{H_t}

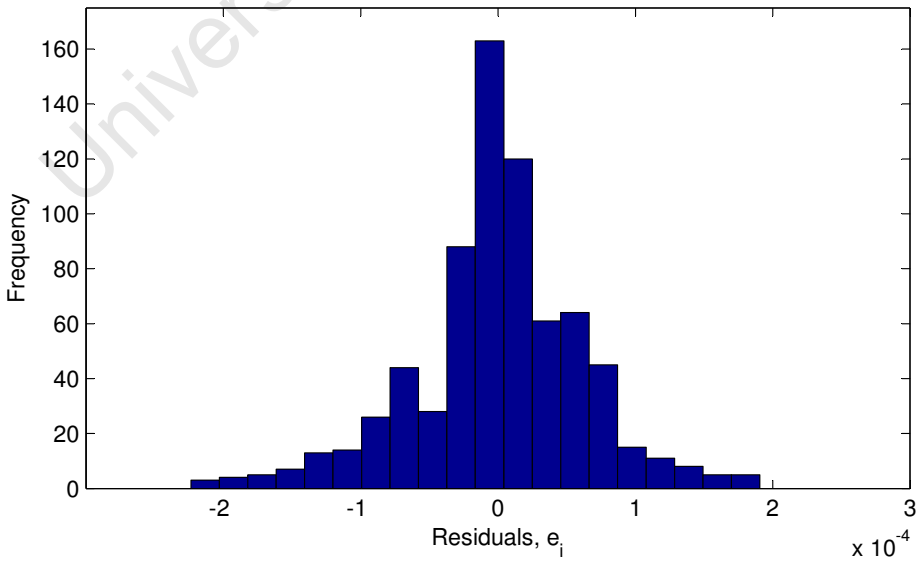


Figure 10.45: Histogram of e_i distribution for C_{H_t}

11 Conclusions

11.1 Aerodynamic Analyses

XFoil promised accurate results for all subsonic conditions, and comparison with CFD results suggests this to be true. All the comparative plots in §5.6 correspond quite well until transonic conditions are reached. A discrepancy is seen with the laminar separation bubbles which only appear on XFoil's results. These are specifically calculated in XFoil and may be present only in low-speed flows. Indeed, the Airbus/Onera study in §3.6.1 investigating laminar separation bubbles was conducted at Mach 0.15, which may suggest that beyond this speed their presence diminishes. XFoil's results of §5.6 show evidence of laminar separation bubbles all the way up to Mach 0.7 (Fig. 5.28), or Mach 0.5 (Fig. 5.26) as the last theoretically valid result. The fully-turbulent flow assumption for airliner wings also suggests that these bubbles would not be present at or near cruise conditions.

The comparative plots of TSFoil and CFD in §5.7 showed a fair correlation when the normal shock assumption was not violated. However, simulations as low as Mach 0.7 violated this assumption for the fine mesh, and the results can be seen to deviate substantially from the CFD solution (Fig. 5.31). Since large normal shocks are generally undesirable it is assumed that this problem might not occur using an accurate airliner airfoil profile. However, TSFoil's poor resolution may not be adequate for the purposes of a further analysis.

CFD demonstrated its ability to model flow separation, and although the shock is distributed over several cells results seem fairly accurate. Some simulations failed to converge but this is more likely to be due to the author's inexperience with CFD than with any shortcomings of the current state of the art of the method. For an experienced user setting up the mesh and generating necessary responses for the L_{81} analysis should take less than 2 weeks, so this proves to be a feasible method to use, especially as the other (potential flow) analyses proved to be insufficient for typical airliner cruise conditions.

11.2 Optimisation

The optimisation procedure performed on the L_{81} test array did not suffer from any great inaccuracies compared with the full-factorial test array, as evidenced by the magnitude of the residual seen in Figs. 10.30, 10.33 and 10.36 compared with Figs. 10.39, 10.42 and 10.45. For all cases the residual was of the order of between

10^{-3} and 10^{-4}. The merit function analysed using equations from both analyses gave near-identical results, which further supports the efficacy of using a reduced array. However, the number of parameters that can be varied is limited, where only 6 factors can be tested without prior knowledge of significant interactions.

Graphical (or additive-model based) investigations of interactions between factors proved to be an inadequate method of predicting significant interaction compared with the numerical ANOVA method. Here significant interactions were detected where it was not graphically obvious that the term should be included (for example Fig. 10.7 shows no obvious interaction yet was detected as significant in the ANOVA analysis of C_L).

Analysis of factor effects of the merit function in §10.5 suggested an optimum airfoil configuration of a thin, cambered airfoil, with x_a moved backwards and x_t moved forward. This result, however, was obtained using PABLO – an inviscid, subsonic solver – so is liable to some error due to the flowfield results being invalid for the required transonic conditions. A similar analysis using more accurate CFD results may yield a different optimum.

The strange situation of finding no aileron deflection that would zero C_{Ha} for a given tab deflection, as explained in §10.4.2, is assumed to be an artifact of the given mathematical model for C_{Ha}. However, it cannot be strictly guaranteed that the tab will always be able to deflect the aileron by the required amount in practice. This dissertation did not attempt to answer this question, and no evidence was uncovered to suggest an answer. This is perhaps an avenue for further research.

The merit function analysed did not include a drag term. In this sense this dissertation does nothing to examine the feasibility of the Flettner-rudder solution from a holistic point of view. If the tab *can* effectively control the aileron, removal of the redundant hydraulics will save weight, but the drag may be increased due to the new rudder configuration. Consideration needs to be given to ensure that any increase in drag is more than compensated for by the reduction in weight. This will depend on how often the ailerons are deflected during an average flight, as an undeflected tab will do little to increase drag compared with the previous system. This might be significant though, for example during flights in high freestream turbulence where small directional changes are constantly being made, or for short flights with long sustained banking manoeuvres.

References

[1] Roskam, J., Lan, C.E., *Airplane Aerodynamics and Performance*, DARcorporation, 3rd Edition, 2003, ISBN 1-884885-44-6.

[2] Anderson, J. D., *Fundamentals of Aerodynamics*, 4th ed., McGraw Hill, 2007, ISBN 007-125408-0.

[3] Sahai, Y., Emi, T., *Tundish Technology for Clean Steel Production*, World Scientific Publishing Co., Singapore, 2008, ISBN 981-270-621-6.

[4] Myers, R.H., Montgomery, D.C., *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, John Wiley & Sons, Inc., 1995, ISBN 0-471-58100-3.

[5] `http://www.symscape.com/reynolds-averaged-navier-stokes-equations`

[6] Turner, T.L., *Thermomechanical Response of Shape Memory Alloy Hybrid Composites*, NASA Langley Research Center, Hampton, Virginia, January 2001, NASA/TM-2001-210656

[7] Hui Min Neo, *More airlines fold as fuel bills soar*, Business Day Online, 9 July 2008, `http://www.iol.co.za/business/international/more-airlines-fold-as-fuel-bills-soar-1.712063`.

[8] Keith Campbell, *Global aviation industry tackling environmental challenges*, Engineering News, 30 May 2008, `http://www.engineeringnews.co.za/article/global-aviation-industry-tackling-environmental-challenges-2008-05-30`.

[9] Flettner, A., *Device for Steering Aircraft*, US Patent Publication, Publication No. US 1574567 A published on 23-Feb-1926, Application No. US 392840 A filed on 29-Jun-1920

[10] Phillips, W.H., *Journey in Aeronautical Research – A Career at NASA Langley Research Center*, Monographs in Aerospace History, Number 12, 1998.

[11] Ursache, N.M, Melin, T., Isikveren, A.T., Friswell, M.I., *Morphing Winglets for Aircraft Multi-phase Improvement*, 7th AIAA Aviation Technology, Integration and Operations Conference (ATIO), 18–20 September 2007, Belfast, Northern Ireland.

[12] Danehy, P.M., DeLoach, R., Cutler, A.D., *Application of Modern Design of Experiments to CARS Thermometry in a Supersonic Combustor*, 22nd AIAA Aerodynamic Measurement Technology and Ground Testing Conference, 2002, AIAA-2002-2914.

[13] DeLoach, R., *Design of Orion Soil Impact Study Using the Modern Design of Experiments*, 48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, 2010, AIAA-2010-0931.

[14] DeLoach, R., Rayos, E.M., Campbell, C.H., Rickman, S.L., Larsen, C.E., *Shuttle Debris Impact Tool Assessment Using the Modern Design of Experiments*, 45th AIAA Aerospace Sciences Meeting and Exhibit, 2007, AIAA-2007-0550.

[15] Rhode, M.N., DeLoach, R., *Hypersonic Wind Tunnel Calibration Using the Modern Design of Experiments*, 41st AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit, 2005, AIAA-2005-4274.

[16] Underwood, P.J., Everhart J.L., DeLoach, R., *National Transonic Facility Wall Pressure Calibration Using Modern Design of Experiments*, 39th AIAA Aerospace Sciences Meeting & Exhibit, 2001, AIAA-2001-0171.

[17] Deloach, R., *Analysis of Wind Tunnel Polar Replicates Using the Modern Design of Experiments*, 27th AIAA Aerodynamic Measurement Technology and Ground Testing Conference, 2010, AIAA-2010-4927.

[18] Katz, A.P., *Low-speed Aerodynamics*, 2nd ed., Cambridge University Press, 2001, ISBN 0-521-66552-3.

[19] Jameson, A., *A perspective on computational algorithms for aerodynamic analysis and design*, Progress in Aerospace Sciences, **37**, 2001, 197–243.

[20] `http://www.intel.com/support/processors/sb/cs-023143.htm`.

[21] Stein, E., De Borst, R., Hughes, T.J.R., *Encyclopedia of Computational Mechanics*, Volume 3: Fluids, John Wiley & Sons, Inc., 2004, ISBN 0-470-84699-2.

[22] Holst, T.L., *Transonic flow computations using nonlinear potential methods*, Progress in Aerospace Sciences, **36**, 2000, 1–61.

[23] Schlichting, H., Gersten, K., *Boundary Layer Theory*, Springer, 8th edition, 2000, ISBN 3-540-66270-7.

[24] White, F.M., *Fluid Mechanics*, 5th Edition, McGraw Hill, 2003, ISBN 0-07-240217-2.

[25] Drela, M., *Assorted views on teaching of aerodynamics*, AIAA, 1998, AIAA-98-2792

[26] Moran, J., *An Introduction to Theoretical and Computational Aerodynamics*, Dover Publications, 2003, ISBN 0-486-42879-6.

[27] Head, M.R., *Entrainment in the Turbulent Boundary Layer*, Ministry Of Aviation, Aeronautical Research Council, Reports And Memoranda, 1960, R. & M. No. 3152.

[28] Green, J.E., Weeks, D.J., Brooman, J.W.F., *Prediction of Turbulent Boundary Layers and Wakes in Compressible Flow by a Lag-Entrainment Method*, Ministry of Defense, Aeronautical Research Council, Reports and Memoranda, 1977, R. & M. No. 3791.

[29] Cebeci, T., Bradshaw, P., *Momentum Transfer in Boundary Layers*, McGraw-Hill, New-York, 1977, ISBN 0-07-010300-3.

[30] Bradshaw, P., Ferriss, D. H., *Calculation of Boundary-Layer Development Using the Turbulent Energy Equation: Compressible Flow on Adiabatic Walls*, Journal of Fluid Mechanics, **46**, 1970, 83–110

[31] Suluksna, K., Juntasaro, E., *Assessment of intermittency transport equations for modeling transition in boundary layers subjected to freestream turbulence*, International Journal of Heat and Fluid Flow, **29**, 2008, 48–61.

[32] Arnal, D., Casalis, G., *Laminar-turbulent transition prediction in three-dimensional flows*, Progress in Aerospace Sciences, **36**, 2000, 173–191.

[33] Gad-el-Hak, M., Tsai, H.M., *Transition and turbulence control*, Volume 8, World Scientific Publishing Co., 2006, ISBN 981-256-594-9

[34] Jiang, L, Choudhari, M., Chang, C.L., Liu, C., *Direct Numerical Simulations of Instability-Wave Generation and Propagation in Supersonic Boundary Layers*, Computational Science and Its Applications - ICCSA 2003, International Conference, Montreal, Canada, May 18-21, 2003, Proceedings, Part II. Published in *Lecture Notes in Computer Science, Vol. 2668*, Springer, ISBN 978-3-540-40161-2.

[35] Chernyshev, S.L., Kiselev, A.P., Kuryachii, A.P., *Laminar flow control research at TsAGI: Past and present*, Progress in Aerospace Sciences, 2010, 169–185.

[36] Johnson, M.W., Ercan, A.H., *A physical model for bypass transition*, International Journal of Heat and Fluid Flow, **20**, 1999, 95–104.

[37] Drela, M., Youngren, H., *XFOIL 6.9 User Primer — THE last update*, 30 Nov 2001, `http://web.mit.edu/drela/Public/web/xfoil/xfoil_doc.txt`.

[38] Van Dam, C.P., *Recent experience with different methods of drag prediction*, Progress in Aerospace Sciences, **35**, 1999, 751–798.

[39] Miley, S.J., van Dam, C.P., Yip, L.P., Willard, P.E., Crowder, J.P., and Watzlavick, R.L., *Slat Transition Characteristics On The NASA B737-100 Aircraft Using Infrared Imaging And Hot-Film Anemometry*, Flow Visualization VII, J. Crowder (Ed.), Begell House, Sep. 1995, 950–956.

[40] Johanson, J., *Prediction of Laminar/Turbulent Transition in Airfoil Flows*, Risø National Laboratory, Roskilde, Denmark, May 1997

[41] Smith, A.M.O., Gamberoni, N., *Transition, pressure gradient and stability theory*, Report ES 26388, Douglas Aircraft Co., El Segundo, California, 1956.

[42] Van Ingen, J.L., *A suggested semi-empirical method for the calculation of boundary layer transition region*, Report UTH-74, University of Technology, Department of Aeronautical Engineering, Delft, 1956.

[43] Windte, J., Scholz, U., Radespiel, R., *Validation of the RANS-simulation of laminar separation bubbles on airfoils*, Aerospace Science and Technology, **10**, 2006, 484–494.

[44] Sandham, N.D., *Transitional separation bubbles and unsteady aspects of aerofoil stall*, The Aeronautical Journal, Volume 112 No. 1133, 395–404.

[45] Gleyzes, C., Patrick Capbern, P., *Experimental study of two AIRBUS/ONERA airfoils in near stall conditions. Part I: Boundary layers*, Aerospace Science and Technology, **7**, 2003, 439–449.

[46] Bendiksen, O.O., *Review of unsteady transonic aerodynamics: Theory and applications*, Progress in Aerospace Sciences, **47**, 2011, 135–167.

[47] `http://www.cfd-online.com/Wiki/Turbulence_modeling`.

[48] *Fluent 6.3 User's Guide*, Fluent Inc., 2006.

[49] Chandler, D., *Record-breaking Daedalus project marks 20th anniversary*, MIT News Office, April 22, 2008, `http://web.mit.edu/newsoffice/2008/daedalus-0422.html`.

[50] `http://www.dfrc.nasa.gov/Gallery/Photo/Daedalus/HTML/EC88-0059-002.html`.

[51] Hoogedoorn, E., Jacobs, G.B., Beyene, A., *Aero-elastic behavior of a flexible blade for wind turbine application: A 2D computational study*, Energy, **35**, 2010, 778–785.

[52] Vermeera, L.J., Sørensen, J.N., Crespo, A., *Wind turbine wake aerodynamics*, Progress in Aerospace Sciences, **39**, 2003, 467–510.

[53] Du, Z., Selig, M.S., *The effect of rotation on the boundary layer of a wind turbine blade*, Renewable Energy, **20**, 2000, 167–181.

[54] Puterbaugh, M, Beyene, A., *Parametric dependence of a morphing wind turbine blade on material elasticity*, Energy, **36**, 2011, 466–474.

[55] Ameku, K., Nagai, B.M., Roy, J.N., *Design of a 3 kW wind turbine generator with thin airfoil blades*, Experimental Thermal and Fluid Science, **32**, 2008, 1723–1730.

[56] Audierne, E., Elizondo, J., Bergami, L.„ Ibarra, H., Probst, O., *Analysis of the furling behavior of small wind turbines*, Applied Energy, **87**, 2010, 2278–2292.

[57] Batten, W.M.J., Bahaj, A.S., Molland, A.F., Chaplin, J.R., *Hydrodynamics of marine current turbines*, Renewable Energy, **31**, 2006, 249–256.

[58] Benini, E., *Significance of blade element theory in performance prediction of marine propellers*, Ocean Engineering, **31**, 2004, 957–974.

[59] Hsin, C.Y., Wu, J.L., Chang, S.F., *Design And Optimization Method For A Two-Dimensional Hydrofoil*, Conference of Global Chinese Scholars on Hydrodynamics, Shanghai, July 11–14, 2006.

[60] Bahaja, A.S., Battena, W.M.J., McCann, G., *Experimental verifications of numerical predictions for the hydrodynamic performance of horizontal axis marine current turbines*, Renewable Energy, **32**, 2007, 2479–2490.

[61] Drela, M., Giles, M.B., *Viscous-Inviscid Analysis of Transonic and Low Reynolds Number Airfoils*, AIAA Journal, Vol. 25, No. 10, 1987, 1347–1355.

[62] Bailey, F.R., Johnson, M.L., Murman, E.M., *Tsfoil - A Computer Code For Two-Dimensional Transonic Calculations, Including Wind-Tunnel Wall Effects And Wave-Drag Evaluation*, in Aerodynamic Analyses Requiring Advanced Computers, Part 2, NASA-SP-347-PT-2, NASA Langley, 1975.

[63] Mason, W.H., *Transonic Aerodynamics of Airfoils and Wings*, Published Online as Chapter 7 of AOE 4124 – Configuration Aerodynamics course notes, `http://www.aoe.vt.edu/~mason/Mason_f/ConfigAero.html`.

[64] Stahara, S.S., *Operational Manual for Two-Dimensional Transonic Code TSFOIL*, NASA Contractor Report CR-3064, 1978.

[65] Wauquiez, C., *Shape Optimization of Low Speed Airfoils using MATLAB and Automatic Differentiation*, LicentiateŠs Thesis, Royal Institute of Technology, Department of Numerical Analysis and Computing Science, 2000, ISRN KTH/NA/R--00/04--SE.

[66] Phadke, M.S., *Quality Engineering Using Robust Design*, P T R Prentice-Hall, Inc., 1989, ISBN 0-13-745167-9.

[67] Peace, G.S., *Taguchi Methods: A Hands-On Approach To Quality Engineering*, Addison-Wesley Publishing Company, 1993, ISBN 0-201-56311-8.

[68] Draper, N.R., Smith, H., *Applied Regression Analysis*, 3rd edition, John Wiley & Sons, Inc., 1998, ISBN 0-471-17082-8.

[69] `http://www.stat.psu.edu/online/development/stat501/data/alcoholarm.txt`

[70] `http://www.stat.psu.edu/online/development/stat501/05model_check/03model_check_rvf.html`

[71] Abbott, I.H., von Doenhof, A.E., *Theory of Wing Sections*, Dover Publications, Inc., 1959, ISBN 486-60586-8.

[72] *Initial Technical Data for ŞControl Surfaces in Confined SpacesŤ Project*, v2.0, Compiled by Sylvain BOYE, Future Projects, April 2008.

Appendix

Appendices A XFoil Ancillary Code

Below is a Matlab `.m` file which can be used to run XFoil from the shell. The variables from lines 25–35 are used to create a text file of appropriate command-line inputs for XFoil. It is passed to XFoil via the "bang" character (`!`) on line 87, which escapes to the shell, runs XFoil, and uses the inputs as if they were entered at the XFoil console. The version of these inputs presented below causes XFoil to output the pressure distribution, in the form of a hard copy of the output screen, `plot.ps`, and two text files, `Cp.data` which contains the pressure distribution, and `bl-params.data` which contains the boundary-layer parameters u_e/u_∞, $\frac{u_e}{u_\infty}$, δ^*, θ and c_f.


```
1:  % This .m file can be used for the running of XFoil in batch mode using
2:  % the inputs given below.
3:
4:  % Flaps are created accordingly. For viscous analysis, set the Reynolds
5:  % number to non-zero. For high Mach and Reynolds numbers, convergence may
6:  % be unlikely - check the output window for any such errors.
7:
8:  % XFoil.exe must be in the same directory as this .m file. All output will
9:  % go to the C:\ root directory. The following files will be found there:
10: % * - plot.ps
11: %     > contains graphical output of Cp distribution
12: % * - bl-params_a<alpha>_m<mach number>.data
13: %     > contains table of boundary-layer parameters (in unicode text format)
14: % * - Cp_a<alpha>_m<mach number>.data
15: %     > contains table of pressure distribution (in unicode text format)
16: % * - Profile.data
17: %     > contains table of xy coordinates of profile (in unicode text format)
18:
19: % Created by Chris Day
20: % Last edited: 2011-04-12
21: close all
22: clear all
23: clc
24:
25: airfoil = '23012';
26: panels = 280;
27: alpha = 4;
28: delta_a = 4;
29: delta_t = -4;
30: x_a = 0.76;
31: y_a = 0.5;
32: x_t = 0.95;
33: y_t = 0.5;
34: Ma = 0.6;
35: Re = 9.938265E+06;
36:
```

```
37:   % ----- generate a text file with appropriate command-prompt entries -----
38:   fid = fopen('XFoil_inputs.dat', 'wt');
39:   fprintf(fid,['naca ' airfoil '\r']); % load this profile
40:   fprintf(fid,'ppar\r'); % enter panelling parameters menu
41:   fprintf(fid,['n ' num2str(panels) '\r']); % change the number of panels
42:                                             % from 140 (default) to 280 (max)
43:   fprintf(fid,'\r'); % return to previous menu
44:   fprintf(fid,'\r'); % return to previous menu
45:
46:   if delta_a ~= 0 || delta_t ~= 0
47:       fprintf(fid,'gdes\r'); % enter the geometry design menu
48:       if delta_a ~= 0
49:           fprintf(fid,'flap\r'); % create a flap
50:           fprintf(fid,[num2str(x_a) '\r']); % hinge location as x/c from LE
51:           fprintf(fid,'999\r'); % change input format to y/t location
52:           fprintf(fid,[num2str(y_a) '\r']); % hinge y/t location
53:           fprintf(fid,[num2str(delta_a) '\r']); % deflection in degrees
54:       end
55:       if delta_t ~= 0
56:           fprintf(fid,'flap\r'); % create a secondary flap
57:           fprintf(fid,[num2str(x_t) '\r']); % hinge location
58:           fprintf(fid,'999\r'); % for y/t location
59:           fprintf(fid,[num2str(y_t) '\r']); % hinge y/t location
60:           fprintf(fid,[num2str(delta_t) '\r']); % deflection in degrees
61:       end
62:       fprintf(fid,'\r'); % return to previous menu
63:       fprintf(fid,'pane\r'); % copy the buffer airfoil to the current airfoil
64:   end
65:   fprintf(fid,'psav\r'); % save the airfoil profile to a text file
66:   fprintf(fid,['C:\\XFoil\\Profile.data\r']); % name of profile data file
67:   fprintf(fid,'\r'); % return to previous menu
68:   fprintf(fid,'oper\r'); % enter the operating points menu
69:   fprintf(fid,'visc\r'); % enter viscous analysis mode
70:   fprintf(fid,[num2str(Re) '\r']); % set the Reynolds number
71:   fprintf(fid,['mach ' num2str(Ma) '\r']); % set the Mach number
72:   fprintf(fid,'cpmi -2\r'); % maximum (negative) value of y-axis
73:   fprintf(fid,['alfa ' num2str(alpha) '\r']); % set angle of attack
74:   fprintf(fid,'!\r'); % continue iterations if not converged
75:   fprintf(fid,'!\r'); % continue iterations if still not converged
76:   fprintf(fid,'cpwr\r'); % save the Cp distribution to a text file
77:   fprintf(fid,['C:\\Cp_a' num2str(alpha) '_m' num2str(Ma*10) '.data\r']);
78:   fprintf(fid,'hard\r'); % create a hard copy of cp plot (creates plot.ps)
79:   fprintf(fid,'dump\r') % save boundary layer parameters to file
80:   fprintf(fid,['C:\\bl-params_a' num2str(alpha) '_m' num2str(Ma*10) '.data\r']);
81:   fprintf(fid,'\r'); % return to previous menu
82:   fprintf(fid,'\r'); % return to previous menu
83:   fprintf(fid,'quit');
84:   fclose(fid);
85:
86:   % ----- send file to XFoil for processing -----
87:   !XFoil.exe<XFoil_inputs.dat
```

Appendices B TSFoil I/O Files

This appendix contains an example input file used and the two output files generated when running TSFoil.

B.1 Input File `n23012.inp`

Presented here is the input file created for TSFoil. The first few lines contain variable definitions, followed by airfoil coordinates. An explanation of each variable is contained in Table B.1.

Table B.1: TSFoil input variables

Variable	Description
`ALPHA`	Angle of attack
`DELTA`	Airfoil thickness ratio
`EMACH`	Mach number
`BCFOIL`	Option for type of airfoil (`3` = xy-coordinates)
`NU`	Number of upper surface coordinates
`NL`	Number of lower surface coordinates
`MAXIT`	Maximum number of iterations on fine mesh
`XU`	Upper surface x-coordinates
`YU`	Upper surface y-coordinates
`XL`	Lower surface x-coordinates
`YL`	Lower surface y-coordinates

```
TEST OF DATA POINT INPUT CASE , NACA23012
 $INP
 ALPHA=4.0,DELTA=0.12,EMACH=0.60
 BCFOIL=3,NU=71,NL=69,MAXIT=1500
 XU = 0, 0.000155, 0.000606, 0.001354, 0.002405,
      0.003767, 0.005455, 0.007497, 0.009934, 0.012822,
      0.016234, 0.020264, 0.025025, 0.030646, 0.037259,
      0.044977, 0.053869, 0.063937, 0.07512, 0.087317,
      0.10042, 0.11434, 0.12901, 0.14441, 0.1605,
      0.1773, 0.19477, 0.21282, 0.23125, 0.24993,
      0.26877, 0.28774, 0.30682, 0.32599, 0.34523,
      0.36455, 0.38392, 0.40335, 0.42282, 0.44234,
      0.46189, 0.48148, 0.5011, 0.52075, 0.54043,
      0.56012, 0.57983, 0.59956, 0.61929, 0.63902,
      0.65874, 0.67839, 0.69786, 0.71691, 0.73506,
```

```
     0.75155, 0.76605, 0.78063, 0.79717, 0.81523,
     0.83407, 0.85323, 0.8724, 0.89132, 0.90955,
     0.92643, 0.94114, 0.95362, 0.96633, 0.97962,
     0.99201,
 YU = 8e-005, 0.002255, 0.004522, 0.006863, 0.009278,
     0.011772, 0.014355, 0.017043, 0.019857, 0.022819,
     0.025953, 0.029282, 0.032824, 0.036584, 0.040546,
     0.044663, 0.048849, 0.052988, 0.056954, 0.060631,
     0.063936, 0.066815, 0.069249, 0.071239, 0.072807,
     0.07399, 0.074845, 0.07544, 0.075813, 0.075975,
     0.075937, 0.075713, 0.075314, 0.074753, 0.074039,
     0.073183, 0.072194, 0.07108, 0.069849, 0.068508,
     0.067063, 0.065521, 0.063888, 0.062168, 0.060366,
     0.058487, 0.056535, 0.054513, 0.052426, 0.050276,
     0.048068, 0.045811, 0.043519, 0.041224, 0.039,
     0.036904, 0.035015, 0.032049, 0.028721, 0.025008,
     0.021092, 0.017059, 0.012971, 0.008885, 0.004895,
     0.00116, -0.002136, -0.004813, -0.006787, -0.008882,
     -0.010865,
 XL = 0, 0.000143, 0.000642, 0.00155, 0.002887,
     0.004644, 0.0068, 0.009344, 0.012289, 0.015677,
     0.019579, 0.024108, 0.029427, 0.035773, 0.043473,
     0.052952, 0.064674, 0.078937, 0.095567, 0.11387,
     0.133, 0.15235, 0.17156, 0.19052, 0.20927,
     0.22797, 0.24673, 0.2656, 0.28457, 0.30364,
     0.32281, 0.34205, 0.36137, 0.38075, 0.4002,
     0.4197, 0.43924, 0.45883, 0.47846, 0.49812,
     0.51782, 0.53754, 0.55729, 0.57706, 0.59685,
     0.61665, 0.63644, 0.65621, 0.67588, 0.6953,
     0.71417, 0.7318, 0.74759, 0.76135, 0.77563,
     0.79224, 0.81053, 0.8297, 0.84925, 0.86884,
     0.8882, 0.9069, 0.92428, 0.93957, 0.95277,
     0.96592, 0.97944, 0.99196, 0.99925,
 YL = 8e-005, -0.002077, -0.00427, -0.006426, -0.008485,
     -0.010402, -0.012165, -0.013783, -0.015277, -0.016672,
     -0.017993, -0.019263, -0.020509, -0.021758, -0.023043,
     -0.024403, -0.025884, -0.027526, -0.029341, -0.031292,
     -0.033308, -0.035303, -0.037198, -0.038923, -0.040416,
     -0.041664, -0.042692, -0.04352, -0.044164, -0.044636,
     -0.044947, -0.045107, -0.045127, -0.045014, -0.044777,
     -0.044423, -0.04396, -0.043394, -0.04273, -0.041975,
     -0.041133, -0.040209, -0.039208, -0.038133, -0.036989,
     -0.035777, -0.034503, -0.033171, -0.031787, -0.030366,
     -0.028936, -0.027544, -0.026313, -0.025136, -0.024916,
     -0.024645, -0.024295, -0.023882, -0.02341, -0.022885,
     -0.022315, -0.021713, -0.021114, -0.020539, -0.019906,
     -0.018393, -0.016864, -0.015413, -0.014556,
 $END
```

B.2 Output File `output.out`

```
 ****************************************************************
 *                                                              *
 *                        PROGRAM TSFOIL                        *
 *                            SOLVES                            *
 *    INVISCID FLOW PAST THIN TWO DIMENSIONAL LIFTING AIRFOIL   *
 *                            USING                             *
 *              TRANSONIC SMALL DISTURBANCE THEORY              *
 *        FULLY CONSERVATIVE FINITE DIFFERENCE EQUATIONS        *
 *                SUCCESSIVE LINE OVERRELAXATION                *
 *                                                              *
 *                         WRITTEN BY                           *
 *                                                              *
 *              EARLL M. MURMAN AND FRANK R. BAILEY             *
 *                 NASA-AMES RESEARCH CENTER                    *
 *                 MOFFETT FIELD, CALIFORNIA                    *
 *                            AND                               *
```

```
    *                      MARGARET L. JOHNSON                       *
    *                  COMPUTER SCIENCES CORPORATION                 *
    *                    MOUNTAIN VIEW, CALIFORNIA                   *
    *                                                                *
    *            Documented in NASA SP-347 and NASA CR-3064          *
    *                                                                *
    *                          Version for                           *
    *               VT Aerospace Design Software Series              *
    *                                                                *
    *               Contact info:                                    *
    *               Dr. William H. Mason                             *
    *               Aerospace & Ocean Engineering, Virginia Tech     *
    *               Blacksburg, VA24061                              *
    *               Email: whmason@vt.edu                            *
    *                                                                *
    ******************************************************************
1    TEST OF DATA POINT INPUT CASE , NACA23012
0    EMACH =  0.60000     POR =  0.00000   IMIN =   1   BCTYPE =  1     AMESH = F
0    DELTA =  0.12000   CLSET =  0.00000  IMAXI =  77   BCFOIL =  3      PHYS = T
0    ALPHA =  4.00000     EPS =  0.20000   JMIN =   1   PSTART =  1     PSAVE = F
0       AK =  0.00000    RIGF =  0.00000  JMAXI =  56   PRTFLO =  1     KUTTA = T
0      GAM =  1.40000   WCIRC =  1.00000  MAXIT =1500   IPRTER = 10      FCR = T
0        F =  0.00000  CVERGE =  0.00001     NU =  71   SIMDEF =  3
0        H =  0.00000  DVERGE =     10.0     NL =  69     ICUT =  2
0       WE = 1.80,1.90,1.95
0    XIN
      -1.075000  -0.950000  -0.825000  -0.700000  -0.575000  -0.450000
      -0.350000  -0.250000  -0.175000  -0.125000  -0.075000  -0.052500
      -0.035000  -0.022500  -0.015000  -0.007500  -0.002500   0.002500
       0.007500   0.012500   0.017500   0.022500   0.027500   0.032500
       0.037500   0.045000   0.055000   0.065000   0.075000   0.085000
       0.097500   0.115000   0.140625   0.171875   0.203125   0.234375
       0.265625   0.296875   0.328125   0.359375   0.390625   0.421875
       0.453125   0.484375   0.515625   0.546875   0.578125   0.609375
       0.640625   0.671875   0.703125   0.734375   0.765625   0.796875
       0.828125   0.859375   0.885000   0.900000   0.915000   0.930000
       0.945000   0.960000   0.975000   0.990000   1.000000   1.010000
       1.025000   1.050000   1.090000   1.150000   1.225000   1.300000
       1.400000   1.500000   1.625000   1.750000   1.875000
0    YIN
      -5.200000  -4.400000  -3.600000  -3.000000  -2.400000  -1.950000
      -1.600000  -1.350000  -1.150000  -0.950000  -0.800000  -0.650000
      -0.550000  -0.450000  -0.390000  -0.340000  -0.300000  -0.270000
      -0.240000  -0.210000  -0.180000  -0.150000  -0.125000  -0.100000
      -0.075000  -0.050000  -0.030000  -0.010000   0.010000   0.030000
       0.050000   0.075000   0.100000   0.125000   0.150000   0.180000
       0.210000   0.240000   0.270000   0.300000   0.340000   0.390000
       0.450000   0.550000   0.650000   0.800000   0.950000   1.150000
       1.350000   1.600000   1.950000   2.400000   3.000000   3.600000
       4.400000   5.200000
0              XU
       0.000000   0.000155   0.000606   0.001354   0.002405   0.003767
       0.005455   0.007497   0.009934   0.012822   0.016234   0.020264
       0.025025   0.030646   0.037259   0.044977   0.053869   0.063937
       0.075120   0.087317   0.100420   0.114340   0.129010   0.144410
       0.160500   0.177300   0.194770   0.212820   0.231250   0.249930
       0.268770   0.287740   0.306820   0.325990   0.345230   0.364550
       0.383920   0.403350   0.422820   0.442340   0.461890   0.481480
       0.501100   0.520750   0.540430   0.560120   0.579830   0.599560
       0.619290   0.639020   0.658740   0.678390   0.697860   0.716910
       0.735060   0.751550   0.766050   0.780630   0.797170   0.815230
       0.834070   0.853230   0.872400   0.891320   0.909550   0.926430
       0.941140   0.953620   0.966330   0.979620   0.992010
0              YU
       0.000080   0.002255   0.004522   0.006863   0.009278   0.011772
       0.014355   0.017043   0.019857   0.022819   0.025953   0.029282
       0.032824   0.036584   0.040546   0.044663   0.048849   0.052988
       0.056954   0.060631   0.063936   0.066815   0.069249   0.071239
       0.072807   0.073990   0.074845   0.075440   0.075813   0.075975
       0.075937   0.075713   0.075314   0.074753   0.074039   0.073183
       0.072194   0.071080   0.069849   0.068508   0.067063   0.065521
       0.063888   0.062168   0.060366   0.058487   0.056535   0.054513
       0.052426   0.050276   0.048068   0.045811   0.043519   0.041224
       0.039000   0.036904   0.035015   0.032049   0.028721   0.025008
       0.021092   0.017059   0.012971   0.008885   0.004895   0.001160
      -0.002136  -0.004813  -0.006787  -0.008882  -0.010865
0              XL
       0.000000   0.000143   0.000642   0.001550   0.002887   0.004644
       0.006800   0.009344   0.012289   0.015677   0.019579   0.024108
       0.029427   0.035773   0.043473   0.052952   0.064674   0.078937
       0.095567   0.113870   0.133000   0.152350   0.171560   0.190520
       0.209270   0.227970   0.246730   0.265600   0.284570   0.303640
       0.322810   0.342050   0.361370   0.380750   0.400200   0.419700
       0.439240   0.458830   0.478460   0.498120   0.517820   0.537540
       0.557290   0.577060   0.596850   0.616650   0.636440   0.656210
       0.675880   0.695300   0.714170   0.731800   0.747590   0.761350
       0.775630   0.792240   0.810530   0.829700   0.849250   0.868840
```

```
      0.888200   0.906900   0.924280   0.939570   0.952770   0.965920
      0.979440   0.991960   0.999250
0             YL
      0.000080  -0.002077  -0.004270  -0.006426  -0.008485  -0.010402
     -0.012165  -0.013783  -0.015277  -0.016672  -0.017993  -0.019263
     -0.020509  -0.021758  -0.023043  -0.024403  -0.025884  -0.027526
     -0.029341  -0.031292  -0.033308  -0.035303  -0.037198  -0.038923
     -0.040416  -0.041664  -0.042692  -0.043520  -0.044164  -0.044636
     -0.044947  -0.045107  -0.045127  -0.045014  -0.044777  -0.044423
     -0.043960  -0.043394  -0.042730  -0.041975  -0.041133  -0.040209
     -0.039208  -0.038133  -0.036989  -0.035777  -0.034503  -0.033171
     -0.031787  -0.030366  -0.028936  -0.027544  -0.026313  -0.025136
     -0.024916  -0.024645  -0.024295  -0.023882  -0.023410  -0.022885
     -0.022315  -0.021713  -0.021114  -0.020539  -0.019906  -0.018393
     -0.016864  -0.015413  -0.014556


         SCALED POR=   0.00000
 AIRFOIL GEOMETRY INFORMATION
 PRINTOUT IN PHYSICAL VARIABLES NORMALIZED BY CHORD LENGTH   MAX THICKNESS =  0.120031
 AIRFOIL VOLUME =  0.08218200                                MAX CAMBER    =  0.018348

                  UPPER  SURFACE              LOWER  SURFACE
       X             Y         DY/DX           Y         DY/DX        THICKNESS    CAMBER
   0.00250000   0.00945611  1.87647808   -0.00806529 -1.15101075     0.00876070  0.00069541
   0.00750000   0.01704670  1.23483181   -0.01264250 -0.66484421     0.01484460  0.00220210
   0.01250000   0.02250460  0.98196942   -0.01537230 -0.44874987     0.01893845  0.00356615
   0.01750000   0.02703708  0.84157652   -0.01732112 -0.33966336     0.02217910  0.00485798
   0.02250000   0.03099201  0.74552447   -0.01883837 -0.27231601     0.02491519  0.00607682
   0.02750000   0.03452988  0.67255974   -0.02008265 -0.22805730     0.02730627  0.00722361
   0.03250000   0.03774063  0.61372125   -0.02114051 -0.19677344     0.02944057  0.00830006
   0.03750000   0.04068227  0.56434661   -0.02206475 -0.17418237     0.03137351  0.00930876
   0.04500000   0.04467455  0.50218338   -0.02327589 -0.15053862     0.03397522  0.01069933
   0.05500000   0.04934381  0.43399227   -0.02467458 -0.13106960     0.03700919  0.01233462
   0.06500000   0.05339198  0.37725285   -0.02592306 -0.11968721     0.03965752  0.01373446
   0.07500000   0.05691463  0.32837617   -0.02708416 -0.11310071     0.04199939  0.01491524
   0.08500000   0.05997987  0.28560391   -0.02819570 -0.10955466     0.04408779  0.01589208
   0.09750000   0.06325262  0.23906589   -0.02954851 -0.10723758     0.04640057  0.01685206
   0.11500000   0.06693702  0.18394616   -0.03141174 -0.10593992     0.04917438  0.01776264
   0.14062500   0.07079805  0.12042949   -0.03410144 -0.10351806     0.05244974  0.01834831
   0.17187500   0.07365213  0.06591441   -0.03722804 -0.09530994     0.05544009  0.01821204
   0.20312500   0.07515178  0.03333523   -0.03995293 -0.07750149     0.05755235  0.01759942
   0.23437500   0.07585476  0.01239813   -0.04203909 -0.05657615     0.05894693  0.01690784
   0.26562500   0.07595658 -0.00540412   -0.04352097 -0.03875849     0.05973878  0.01621781
   0.29687500   0.07554273 -0.02076648   -0.04448781 -0.02344138     0.06001527  0.01552746
   0.32812500   0.07468101 -0.03415640   -0.04500587 -0.00999709     0.05984344  0.01483757
   0.35937500   0.07342541 -0.04592589   -0.04513122  0.00175662     0.05927832  0.01414710
   0.39062500   0.07182298 -0.05640172   -0.04490862  0.01224618     0.05836580  0.01345718
   0.42187500   0.06991126 -0.06575165   -0.04437668  0.02160490     0.05714397  0.01276729
   0.45312500   0.06772296 -0.07419388   -0.04356906  0.02995584     0.05564601  0.01207695
   0.48437500   0.06528553 -0.08167113   -0.04251205  0.03752403     0.05389879  0.01138674
   0.51562500   0.06262443 -0.08853099   -0.04123094  0.04438299     0.05192768  0.01069675
   0.54687500   0.05975911 -0.09479306   -0.03974514  0.05058086     0.04975212  0.01000699
   0.57812500   0.05670658 -0.10048467   -0.03807314  0.05629309     0.04738986  0.00931672
   0.60937500   0.05348281 -0.10577126   -0.03622989  0.06166301     0.04485635  0.00862646
   0.64062500   0.05009848 -0.11073159   -0.03422592  0.06652380     0.04216220  0.00793628
   0.67187500   0.04656527 -0.11533500   -0.03207380  0.07128035     0.03931954  0.00724573
   0.70312500   0.04288744 -0.12035060   -0.02978374  0.07505175     0.03633559  0.00655185
   0.73437500   0.03908720 -0.12700279   -0.02735373  0.07324248     0.03322047  0.00586674
   0.76562500   0.03508538 -0.16380906   -0.02498375  0.02146427     0.03003457  0.00505081
   0.79687500   0.02877991 -0.19956683   -0.02455315  0.01923630     0.02666653  0.00211338
   0.82812500   0.02233016 -0.20786113   -0.02391810  0.02283086     0.02312413 -0.00079397
   0.85937500   0.01575397 -0.21277975   -0.02314510  0.02686573     0.01944953 -0.00369557
   0.88499999   0.01025596 -0.21642613   -0.02241331  0.03044196     0.01633464 -0.00607867
   0.89999998   0.00699167 -0.21881256   -0.02193885  0.03253773     0.01446526 -0.00747359
   0.91500002   0.00369248 -0.22099784   -0.02144378  0.03394098     0.01256813 -0.00887565
   0.93000001   0.00036669 -0.22251080   -0.02089377  0.03889457     0.01063023 -0.01026354
   0.94499999  -0.00301386 -0.22542363   -0.02035188  0.03879144     0.00866901 -0.01168287
   0.95999998  -0.00585963 -0.14991398   -0.01913395  0.12292326     0.00663716 -0.01249679
   0.97500002  -0.00813718 -0.16062073   -0.01735443  0.10995871     0.00460862 -0.01274580
   0.99000001  -0.01054360 -0.15978540   -0.01564371  0.11767347     0.00255006 -0.01309365
   1.00000000  -0.01214379 -0.16004822   -0.01446783  0.11755825     0.00116202 -0.01330581
1
   WE =  1.8000     EPS =   0.2000     MAXIT FOR THIS MESH =  375

  ITER      CL        CM     IERR JERR    ERROR    IRL  JRL    BIGRL        ERCIRC
    10   0.80228  -0.10011    12    7   0.6157E-01   6    7   0.2018E+03   0.6351E-01
    20   0.93678  -0.06309     5    4   0.2336E-01   6    7   0.4536E+03   0.1077E-01
    30   1.00114  -0.05571    12    7   0.1119E-01   6    7   0.2237E+03   0.7726E-02
    40   1.04464  -0.04851     8    8   0.7298E-02   6    7   0.1198E+03   0.5028E-02
    50   1.06430  -0.04748     5    5   0.3350E-02   6    7   0.6625E+02   0.1620E-02
    60   1.07443  -0.04697    12    7   0.1825E-02   6    7   0.3630E+02   0.5748E-03
    70   1.07973  -0.04689     6    7   0.1007E-02   6    7   0.2012E+02   0.3538E-03
    80   1.08263  -0.04684     6    7   0.5594E-03   6    7   0.1116E+02   0.2726E-03
    90   1.08424  -0.04678     5    6   0.3116E-03   6    7   0.6193E+01   0.1732E-03
   100   1.08514  -0.04675     5    7   0.1732E-03   6    7   0.3434E+01   0.9978E-04
   110   1.08563  -0.04673     5    6   0.9618E-04   6    7   0.1908E+01   0.5555E-04
```

```
  120   1.08591  -0.04672    5    6   0.5337E-04   6   7   0.1059E+01   0.3099E-04
  130   1.08606  -0.04671    5    6   0.2979E-04   6   7   0.5868E+00   0.1740E-04
  140   1.08614  -0.04671    5    6   0.1651E-04   6   7   0.3250E+00   0.9298E-05
  150   1.08619  -0.04671    5    6   0.9256E-05   6   7   0.1803E+00   0.5364E-05


                    ........SOLUTION CONVERGED........
1 FORCE COEFFICIENTS, PRESSURE COEFFICIENT, AND MACH NUMBER
  (OR SIMILARITY PARAMETER) ON BODY AND DIVIDING STREAM LINE.
                    COARSE MESH
0         CL =  1.086190
          CM = -0.046706
         CP* = -1.303869
0                           LOWER                       UPPER
                            Y=0-                        Y=0+
   I        X          CP          M1              CP          M1

   1   -1.075000    0.005437    0.597772        0.005437    0.597772                          B                *
   2   -0.575000    0.048135    0.579977        0.048135    0.579977                         B                 *
   3   -0.175000    0.137736    0.540734        0.137736    0.540734                        B                  *
   4   -0.035000    0.261689    0.481197        0.261689    0.481197                      B                    *
   5   -0.002500    0.333166    0.443245        0.333166    0.443245                     B                     *
                        AIRFOIL LEADING EDGE                                    AIRFOIL LEADING EDGE
   6    0.017500    1.092427    0.000000       -0.318286    0.718491               L                  U       *
   7    0.037500    0.872277    0.000000       -0.867739    0.886525                 L                     U  *
   8    0.075000    0.670383    0.175911       -1.692635    1.091249                    L                     *  U
   9    0.140625    0.423955    0.389748       -1.976007    1.153220                       L                  *     U
  10    0.265625    0.209329    0.507199       -1.267928    0.991140                       L                  *
  11    0.390625    0.123659    0.547085       -0.766572    0.858061                       L           U      *
  12    0.515625    0.123312    0.547241       -0.627860    0.817425                       L         U        *
  13    0.640625    0.179297    0.521529       -0.512205    0.781930                       L         U        *
  14    0.765625    0.191473    0.515767       -0.364358    0.734060                       L       U          *
  15    0.885000    0.150055    0.535113       -0.128474    0.650432                        L   U             *
  16    0.945000    0.135426    0.541781        0.106530    0.554716                        B                 *
  17    1.000000    0.232259    0.495981        0.274083    0.474834                       UL                 *
                        AIRFOIL TRAILING EDGE                                   AIRFOIL TRAILING EDGE
  18    1.090000    0.182825    0.519866        0.182825    0.519866                       B                  *
  19    1.400000    0.087349    0.563139        0.087349    0.563139                        B                 *
  20    1.875000    0.031744    0.586872        0.031744    0.586872                         B                *
0

         Y(J) J=  1 TO 14
         -5.200000   -2.400000   -1.150000   -0.550000   -0.300000   -0.180000
         -0.075000    0.075000    0.180000    0.300000    0.550000    1.150000
          2.400000    5.200000
1
   WE =  1.9000     EPS =   0.2000     MAXIT FOR THIS MESH =  750

  ITER      CL         CM     IERR JERR      ERROR     IRL  JRL     BIGRL        ERCIRC
   10   1.09073  -0.04182    9    12    0.2131E-02    9   12   0.1188E+03   0.5965E-03
   20   1.09246  -0.04207   17    14    0.7514E-03   10   14   0.4267E+02   0.2412E-03
   30   1.09339  -0.04203    7     7    0.4149E-03   10   14   0.3364E+02   0.3372E-03
   40   1.09479  -0.04186   10    15    0.2678E-03   10   14   0.2295E+02   0.1153E-03
   50   1.09556  -0.04185   10    15    0.1960E-03   10   14   0.1644E+02   0.1173E-03
   60   1.09616  -0.04182    9    14    0.1367E-03   10   14   0.1205E+02   0.6831E-04
   70   1.09657  -0.04178    9    14    0.1013E-03   10   14   0.8925E+01   0.5257E-04
   80   1.09690  -0.04176    8    14    0.7403E-04   10   14   0.6491E+01   0.4303E-04
   90   1.09715  -0.04174    9    15    0.5407E-04   10   14   0.4761E+01   0.3111E-04
  100   1.09734  -0.04173   10    14    0.4007E-04   10   14   0.3538E+01   0.2384E-04
  110   1.09748  -0.04172    8    14    0.2979E-04   10   14   0.2618E+01   0.1824E-04
  120   1.09759  -0.04171    9    12    0.2227E-04   10   14   0.1951E+01   0.1395E-04
  130   1.09768  -0.04171    9    14    0.1684E-04   10   14   0.1452E+01   0.1061E-04
  140   1.09774  -0.04170    9    13    0.1265E-04   10   14   0.1091E+01   0.8106E-05
  150   1.09779  -0.04170   10    14    0.9463E-05   10   14   0.8359E+00   0.5960E-05


                    ........SOLUTION CONVERGED........
1 FORCE COEFFICIENTS, PRESSURE COEFFICIENT, AND MACH NUMBER
  (OR SIMILARITY PARAMETER) ON BODY AND DIVIDING STREAM LINE.
                    MEDIUM MESH
0         CL =  1.097788
          CM = -0.041701
         CP* = -1.303869
0                           LOWER                       UPPER
                            Y=0-                        Y=0+
   I        X          CP          M1              CP          M1

   1   -1.075000    0.028664    0.588158        0.028664    0.588158                          B                *
   2   -0.825000    0.041138    0.582930        0.041138    0.582930                         B                 *
   3   -0.575000    0.061412    0.574331        0.061412    0.574331                         B                 *
   4   -0.350000    0.106567    0.554700        0.106567    0.554700                         B                 *
   5   -0.175000    0.201537    0.510956        0.201537    0.510956                       B                   *
   6   -0.075000    0.353645    0.431757        0.353645    0.431757                     B                     *
   7   -0.035000    0.506300    0.333892        0.506300    0.333892                   B                       *
   8   -0.015000    0.614176    0.241937        0.614176    0.241937                  B                        *
   9   -0.002500    0.690303    0.145489        0.690303    0.145489                 B                         *
                        AIRFOIL LEADING EDGE                                    AIRFOIL LEADING EDGE
```

```
  10    0.007500    1.503495    0.000000        0.004448    0.598178           L                 U              *
  11    0.017500    1.181562    0.000000       -0.599313    0.808808              L                      U       *
  12    0.027500    0.999346    0.000000       -0.975235    0.915801                 L                       U   *
  13    0.037500    0.860099    0.000000       -1.416700    1.027318                    L                        * U
  14    0.055000    0.735154    0.000000       -1.791641    1.113293                     L                       *      U
  15    0.075000    0.633809    0.221126       -1.944543    1.146505                       L                     *        U
  16    0.097500    0.532613    0.313956       -2.104019    1.180149                       L                     *          U
  17    0.140625    0.395310    0.407386       -2.210456    1.202080                         L                   *          U
  18    0.203125    0.248040    0.488109       -2.055314    1.169976                           L                 *        U
  19    0.265625    0.151862    0.534284       -1.325272    1.005239                           L                 *
  20    0.328125    0.109246    0.553513       -0.777516    0.861186                             L         U     *
  21    0.390625    0.092564    0.560861       -0.759326    0.855986                             L         U     *
  22    0.453125    0.091175    0.561469       -0.692000    0.836460                             L        U      *
  23    0.515625    0.100774    0.557257       -0.619217    0.814826                             L       U       *
  24    0.578125    0.120546    0.548480       -0.551711    0.794233                            L       U        *
  25    0.640625    0.154797    0.532934       -0.495287    0.776602                            L       U        *
  26    0.703125    0.222118    0.500973       -0.466904    0.767579                            L       U        *
  27    0.765625    0.234382    0.494929       -0.407421    0.748319                            L      U         *
  28    0.828125    0.183583    0.519508       -0.250650    0.695004                             L    U          *
  29    0.885000    0.160418    0.530339       -0.077891    0.631056                             L  U            *
  30    0.915000    0.149948    0.535162        0.052094    0.578299                             LU              *
  31    0.945000    0.137134    0.541007        0.189179    0.516858                             B               *
  32    0.975000    0.198277    0.512520        0.293386    0.464750                            UL               *
  33    1.000000    0.345629    0.436290        0.372776    0.420742                           B                 *
                        AIRFOIL TRAILING EDGE                                          AIRFOIL TRAILING EDGE
  34    1.025000    0.283664    0.469856        0.283664    0.469856                            B                *
  35    1.090000    0.184171    0.519231        0.184171    0.519231                             B               *
  36    1.225000    0.112738    0.551963        0.112738    0.551963                             B               *
  37    1.400000    0.073982    0.568934        0.073982    0.568934                             B               *
  38    1.625000    0.053007    0.577912        0.053007    0.577912                             B               *
  39    1.875000    0.038354    0.584101        0.038354    0.584101                             B               *
0

        Y(J) J=  1 TO 28
        -5.200000   -3.600000   -2.400000   -1.600000   -1.150000   -0.800000
        -0.550000   -0.390000   -0.300000   -0.240000   -0.180000   -0.125000
        -0.075000   -0.030000    0.030000    0.075000    0.125000    0.180000
         0.240000    0.300000    0.390000    0.550000    0.800000    1.150000
         1.600000    2.400000    3.600000    5.200000
1
   WE =  1.9500     EPS =   0.2000     MAXIT FOR THIS MESH = 1500

  ITER     CL        CM     IERR JERR    ERROR    IRL  JRL    BIGRL        ERCIRC
   10   1.09461  -0.03417    9   29   0.1979E-02  23   25   0.4041E+03   0.5153E-03
   20   1.09216  -0.03327   17   22   0.8335E-03  17   22   0.2145E+03   0.3190E-03
   30   1.09044  -0.03284   21   17   0.3518E-03  17   28   0.1304E+03   0.1792E-03
   40   1.08976  -0.03267   13   15   0.2261E-03  17   28   0.7894E+02   0.7534E-04
   50   1.08929  -0.03276   16   33   0.1435E-03  17   28   0.4806E+02   0.5090E-04
   60   1.08910  -0.03283   19   37   0.8783E-04  17   28   0.2458E+02   0.1431E-05
   70   1.08914  -0.03290   11   45   0.6804E-04  17   28   0.1425E+02   0.5126E-05
   80   1.08909  -0.03294   13   46   0.4145E-04  17   28   0.7837E+01   0.3934E-05
   90   1.08910  -0.03297   14   50   0.2412E-04  17   28   0.5593E+01   0.2742E-05
  100   1.08907  -0.03297   16   51   0.1935E-04  17   28   0.4596E+01   0.6199E-05
  110   1.08903  -0.03298   15   49   0.1298E-04  17   28   0.3630E+01   0.4053E-05
  120   1.08902  -0.03298   16   50   0.8999E-05  17   28   0.3201E+01   0.1550E-05


                .........SOLUTION CONVERGED.........
1
0 PRINTOUT IN  PHYSICAL  VARIABLES.
0 DEFINITION OF SIMILARITY PARAMETERS BY KRUPP
0 BOUNDARY CONDITION FOR FREE AIR
0 DIFFERENCE EQUATIONS ARE FULLY CONSERVATIVE.
0 KUTTA CONDITION IS ENFORCED.
0
              MACH =   0.6000000
             DELTA =   0.1200000
             ALPHA =   4.0000000
                 K =   4.3843770
  DOUBLET STRENGTH =   0.6848500
0        PARAMETERS USED TO TRANSFORM VARIABLES
                  TO TRANSONIC SCALING

            CPFACT =   0.3568678
            CDFACT =   0.0428241
            CMFACT =   0.3568678
            CLFACT =   0.3568678
             YFACT =   2.6173630
             VFACT =   6.8754935
1 FORCE COEFFICIENTS, PRESSURE COEFFICIENT, AND MACH NUMBER
  (OR SIMILARITY PARAMETER) ON BODY AND DIVIDING STREAM LINE.
                     FINAL MESH
0         CL =  1.089016
          CM = -0.032978
         CP* = -1.303869
0                           LOWER                       UPPER
                            Y=0-                        Y=0+
```

I	X	CP	M1	CP	M1
1	-1.075000	0.039474	0.583630	0.039474	0.583630
2	-0.950000	0.043764	0.581823	0.043764	0.581823
3	-0.825000	0.049335	0.579469	0.049335	0.579469
4	-0.700000	0.058060	0.575762	0.058060	0.575762
5	-0.575000	0.071712	0.569913	0.071712	0.569913
6	-0.450000	0.092614	0.560839	0.092614	0.560839
7	-0.350000	0.123428	0.547189	0.123428	0.547189
8	-0.250000	0.170775	0.525524	0.170775	0.525524
9	-0.175000	0.237540	0.493360	0.237540	0.493360
10	-0.125000	0.326457	0.446945	0.326457	0.446945
11	-0.075000	0.452258	0.371497	0.452258	0.371497
12	-0.052500	0.592417	0.263086	0.592417	0.263086
13	-0.035000	0.733868	0.000000	0.733868	0.000000
14	-0.022500	0.881137	0.000000	0.881137	0.000000
15	-0.015000	1.017718	0.000000	1.017718	0.000000
16	-0.007500	1.149053	0.000000	1.149053	0.000000
17	-0.002500	1.273296	0.000000	1.273296	0.000000
		AIRFOIL LEADING EDGE			
18	0.002500	2.076996	0.000000	0.504918	0.334907
19	0.007500	1.569385	0.000000	-0.307470	0.714787
20	0.012500	1.266065	0.000000	-0.792978	0.865581
21	0.017500	1.070432	0.000000	-1.116543	0.952917
22	0.022500	0.936948	0.000000	-1.342548	1.009448
23	0.027500	0.843135	0.000000	-1.561090	1.061252
24	0.032500	0.775352	0.000000	-1.702931	1.093562
25	0.037500	0.719573	0.082462	-1.747454	1.103509
26	0.045000	0.666200	0.181653	-1.821590	1.119876
27	0.055000	0.619271	0.236713	-1.917025	1.140599
28	0.065000	0.582244	0.272410	-2.005773	1.159538
29	0.075000	0.551565	0.298774	-2.076433	1.174398
30	0.085000	0.521776	0.322317	-2.135530	1.186684
31	0.097500	0.485961	0.348522	-2.194791	1.198877
32	0.115000	0.435419	0.382460	-2.259481	1.212048
33	0.140625	0.364283	0.425667	-2.314147	1.223067
34	0.171875	0.279841	0.471848	-2.329975	1.226238
35	0.203125	0.204052	0.509747	-2.299869	1.220198
36	0.234375	0.152176	0.534139	-2.232246	1.206520
37	0.265625	0.119833	0.548799	-1.617136	1.074135
38	0.296875	0.098552	0.558235	-0.824836	0.874567
39	0.328125	0.084734	0.564277	-0.640939	0.821342
40	0.359375	0.076340	0.567916	-0.686268	0.834777
41	0.390625	0.072045	0.569769	-0.691139	0.836208
42	0.421875	0.071132	0.570163	-0.676092	0.831780
43	0.453125	0.072933	0.569386	-0.651196	0.824401
44	0.484375	0.077081	0.567596	-0.621835	0.815614
45	0.515625	0.083460	0.564831	-0.590839	0.806233
46	0.546875	0.091938	0.561135	-0.559431	0.796615
47	0.578125	0.102613	0.556446	-0.528741	0.787103
48	0.609375	0.116179	0.550431	-0.499665	0.777984
49	0.640625	0.133593	0.542610	-0.473114	0.769563
50	0.671875	0.157605	0.531639	-0.451498	0.762638
51	0.703125	0.194784	0.514189	-0.439599	0.758799
52	0.734375	0.255658	0.484263	-0.452842	0.763070
53	0.765625	0.259107	0.482512	-0.439544	0.758781
54	0.796875	0.204822	0.509376	-0.342775	0.726808
55	0.828125	0.174276	0.523886	-0.222967	0.685159
56	0.859375	0.157063	0.531889	-0.126109	0.649538
57	0.885000	0.146563	0.536712	-0.047639	0.619180
58	0.900000	0.138922	0.540195	0.014262	0.594137
59	0.915000	0.128390	0.544959	0.075025	0.568484
60	0.930000	0.112802	0.551935	0.149046	0.535576
61	0.945000	0.075977	0.568073	0.251954	0.486137
62	0.960000	0.120688	0.548416	0.297987	0.462314
63	0.975000	0.236869	0.493694	0.305167	0.458486
64	0.990000	0.334958	0.442252	0.367098	0.424042
65	1.000000	0.449295	0.373450	0.457150	0.368252
		AIRFOIL TRAILING EDGE			
66	1.010000	0.395919	0.407019	0.395919	0.407019
67	1.025000	0.314673	0.453369	0.314673	0.453369
68	1.050000	0.242689	0.490792	0.242689	0.490792
69	1.090000	0.183682	0.519461	0.183682	0.519461
70	1.150000	0.138859	0.540224	0.138859	0.540224
71	1.225000	0.107980	0.554074	0.107980	0.554074
72	1.300000	0.086633	0.563450	0.086633	0.563450
73	1.400000	0.070667	0.570363	0.070667	0.570363
74	1.500000	0.058764	0.575462	0.058764	0.575462
75	1.625000	0.049791	0.579276	0.049791	0.579276
76	1.750000	0.043405	0.581975	0.043405	0.581975
77	1.875000	0.038234	0.584152	0.038234	0.584152

0

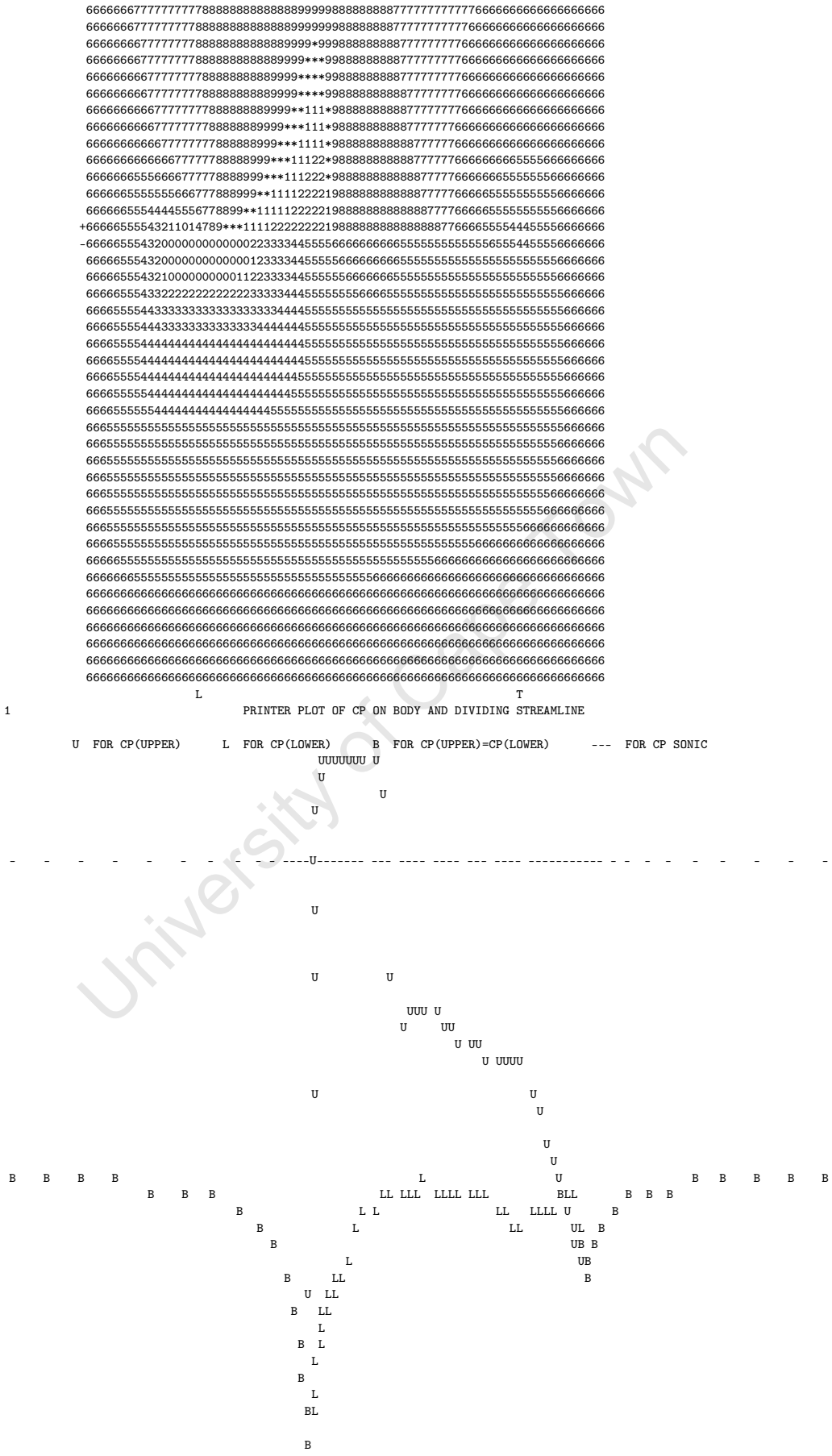
Y(J) J= 1 TO 56

-5.200000 -4.400000 -3.600000 -3.000000 -2.400000 -1.950000
-1.600000 -1.350000 -1.150000 -0.950000 -0.800000 -0.650000
-0.550000 -0.450000 -0.390000 -0.340000 -0.300000 -0.270000

```
        -0.240000   -0.210000   -0.180000   -0.150000   -0.125000   -0.100000
        -0.075000   -0.050000   -0.030000   -0.010000    0.010000    0.030000
         0.050000    0.075000    0.100000    0.125000    0.150000    0.180000
         0.210000    0.240000    0.270000    0.300000    0.340000    0.390000
         0.450000    0.550000    0.650000    0.800000    0.950000    1.150000
         1.350000    1.600000    1.950000    2.400000    3.000000    3.600000
         4.400000    5.200000
1 FLOW AT EACH GRID POINT.  P PARABOLIC
                            H HYPERBOLIC
                            S SHOCK
                            - ELLIPTIC


          56     ------------------------------------------------------------------------
          55     ------------------------------------------------------------------------
          54     ------------------------------------------------------------------------
          53     ------------------------------------------------------------------------
          52     ------------------------------------------------------------------------
          51     ------------------------------------------------------------------------
          50     ------------------------------------------------------------------------
          49     ------------------------------------------------------------------------
          48     ------------------------------------------------------------------------
          47     ------------------------------------------------------------------------
          46     ------------------------------------------------------------------------
          45     ------------------------------------------------------------------------
          44     ------------------------------------------------------------------------
          43     ------------------------------------------------------------------------
          42     ------------------------------------------------------------------------
          41     ------------------------------------------------------------------------
          40     ------------------------------------------------------------------------
          39     ------------------------------------------------------------------------
          38     --------------------------------PS--------------------------------------
          37     ------------------------------PHHS--------------------------------------
          36     ----------------------------PHHHS---------------------------------------
          35     --------------------------PHHHHS-----------------------------------------
          34     ------------------------PHHHHHS-------------------------------------------
          33     ----------------------PHHHHHHS---------------------------------------------
          32     --------------------PHHHHHHHS-----------------------------------------------
          31     ------------------PHHHHHHHHS-------------------------------------------------
          30     ----------------PHHHHHHHHHS---------------------------------------------------
          29     --------------PHHHHHHHHHHHS-----------------------------------------------------
          28     ------------------------------------------------------------------------
          27     ------------------------------------------------------------------------
          26     ------------------------------------------------------------------------
          25     ------------------------------------------------------------------------
          24     ------------------------------------------------------------------------
          23     ------------------------------------------------------------------------
          22     ------------------------------------------------------------------------
          21     ------------------------------------------------------------------------
          20     ------------------------------------------------------------------------
          19     ------------------------------------------------------------------------
          18     ------------------------------------------------------------------------
          17     ------------------------------------------------------------------------
          16     ------------------------------------------------------------------------
          15     ------------------------------------------------------------------------
          14     ------------------------------------------------------------------------
          13     ------------------------------------------------------------------------
          12     ------------------------------------------------------------------------
          11     ------------------------------------------------------------------------
          10     ------------------------------------------------------------------------
           9     ------------------------------------------------------------------------
           8     ------------------------------------------------------------------------
           7     ------------------------------------------------------------------------
           6     ------------------------------------------------------------------------
           5     ------------------------------------------------------------------------
           4     ------------------------------------------------------------------------
           3     ------------------------------------------------------------------------
           2     ------------------------------------------------------------------------
           1     ------------------------------------------------------------------------
1  MACH NO. MAP.   ROUNDED TO NEAREST .1


          66666666666666666666666666666666666666666666666666666666666666666666666666
          66666666666666666666666666666666666666666666666666666666666666666666666666
          66666666666666666666666666666666666666666666666666666666666666666666666666
          66666666666666666666666666666666666666666666666666666666666666666666666666
          66666666666666666666666666666666666666666666666666666666666666666666666666
          66666666666666666666666677777777777777777766666666666666666666666666666666
          66666677777777777777777777777777777777777777777777766666666666666666666666
          66666777777777777777777777777777777777777777777777776666666666666666666666
          66666777777777777777777777777777777777777777777777766666666666666666666666
          66666777777777777777777777777777777777777777777776666666666666666666666666
          66666777777777777777777777777777777777777777777766666666666666666666666666
          66666677777777777777777777777888888888877777777777777777777766666666666666
          66666677777777777777777888888888888888877777777777777777766666666666666666
          66666677777777777788888888888888888888877777777777777766666666666666666666
```

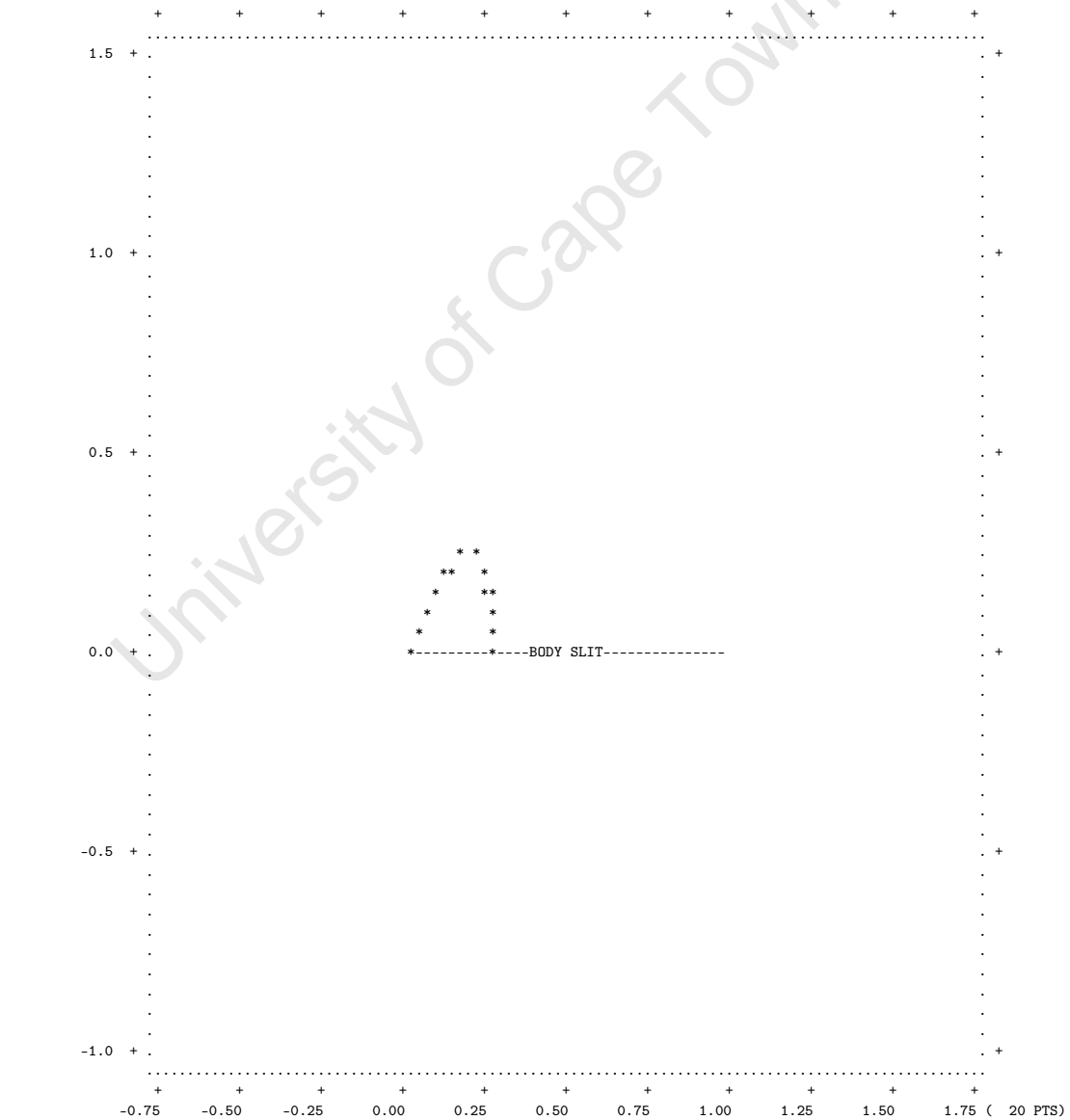


```
                                    BL


                                     L


                                    L
1SONIC LINE COORDINATES
      Y          XSONIC


   0.24000   0.18040   0.22285
   0.21000   0.15173   0.24451
   0.18000   0.13065   0.25445
   0.15000   0.11232   0.26157
   0.12500   0.09901   0.26616
   0.10000   0.08580   0.26987
   0.07500   0.07213   0.27274
   0.05000   0.05631   0.27493
   0.03000   0.04285   0.27631
   0.01000   0.02671   0.27744
  BODY LOCATION
1
                              SONIC LINE PLOT          Y VS X          *  FOR SONIC POINTS
```




```
1
1INVISCID WAKE PROFILES FOR INDIVIDUAL SHOCK WAVES WITHIN MOMENTUM CONTOUR
0SHOCK  1
 WAVE DRAG FOR THIS SHOCK=    0.008083
```

```
      Y         CD(Y)        PO/POINF

   0.00000000  0.07253284  0.95531410
   0.01000000  0.07049683  0.95656842
   0.03000000  0.06653986  0.95900625
   0.05000000  0.06038446  0.96279842
   0.07500000  0.05050071  0.96888763
   0.10000000  0.03953423  0.97564381
   0.12500000  0.02890096  0.98219472
   0.15000001  0.01776557  0.98905504
   0.18000001  0.01078558  0.99335521
   0.20999999  0.00606339  0.99626446
   0.23999998  0.00036787  0.99977338
1CALCULATION OF DRAG COEFFICIENT BY MOMENTUM INTEGRAL METHOD
0BOUNDARIES OF CONTOUR USED               CONTRIBUTION TO CD
 UPSTREAM    X =   -0.175000               CDUP    =   -0.001340
 DOWNSTREAM  X =    1.225000               CDDOWN =    0.003804
 TOP         Y =    4.400000               CDTOP   =    0.000090
 BOTTOM      Y =   -4.400000               CDBOT   =    0.000045
               TOTAL CONTRIBUTIONS AROUND CONTOUR =    0.002599
0THERE ARE  1 SHOCKS INSIDE CONTOUR. TOTAL CDWAVE =    0.008083
0NOTE - ALL SHOCKS CONTAINED WITHIN CONTOUR
 CDWAVE EQUALS TOTAL WAVE DRAG
0DRAG CALCULATED FROM MOMENTUM INTEGRAL    CD     =    0.010682
```

B.3 Output File `fort.1`

```
 TSFOIL2   Mach =   0.600   CL =   1.089
   i      X/C        Cp-up     M-up      Cp-low    M-low
   1   0.0025      0.50492   0.3349      2.07700   0.0000
   2   0.0075     -0.30747   0.7148      1.56939   0.0000
   3   0.0125     -0.79298   0.8656      1.26607   0.0000
   4   0.0175     -1.11654   0.9529      1.07043   0.0000
   5   0.0225     -1.34255   1.0094      0.93695   0.0000
   6   0.0275     -1.56109   1.0613      0.84313   0.0000
   7   0.0325     -1.70293   1.0936      0.77535   0.0000
   8   0.0375     -1.74745   1.1035      0.71957   0.0825
   9   0.0450     -1.82159   1.1199      0.66620   0.1817
  10   0.0550     -1.91702   1.1406      0.61927   0.2367
  11   0.0650     -2.00577   1.1595      0.58224   0.2724
  12   0.0750     -2.07643   1.1744      0.55156   0.2988
  13   0.0850     -2.13553   1.1867      0.52178   0.3223
  14   0.0975     -2.19479   1.1989      0.48596   0.3485
  15   0.1150     -2.25948   1.2120      0.43542   0.3825
  16   0.1406     -2.31415   1.2231      0.36428   0.4257
  17   0.1719     -2.32998   1.2262      0.27984   0.4718
  18   0.2031     -2.29987   1.2202      0.20405   0.5097
  19   0.2344     -2.23225   1.2065      0.15218   0.5341
  20   0.2656     -1.61714   1.0741      0.11983   0.5488
  21   0.2969     -0.82484   0.8746      0.09855   0.5582
  22   0.3281     -0.64094   0.8213      0.08473   0.5643
  23   0.3594     -0.68627   0.8348      0.07634   0.5679
  24   0.3906     -0.69114   0.8362      0.07204   0.5698
  25   0.4219     -0.67609   0.8318      0.07113   0.5702
  26   0.4531     -0.65120   0.8244      0.07293   0.5694
  27   0.4844     -0.62183   0.8156      0.07708   0.5676
  28   0.5156     -0.59084   0.8062      0.08346   0.5648
  29   0.5469     -0.55943   0.7966      0.09194   0.5611
  30   0.5781     -0.52874   0.7871      0.10261   0.5564
  31   0.6094     -0.49967   0.7780      0.11618   0.5504
  32   0.6406     -0.47311   0.7696      0.13359   0.5426
  33   0.6719     -0.45150   0.7626      0.15760   0.5316
  34   0.7031     -0.43960   0.7588      0.19478   0.5142
  35   0.7344     -0.45284   0.7631      0.25566   0.4843
  36   0.7656     -0.43954   0.7588      0.25911   0.4825
  37   0.7969     -0.34278   0.7268      0.20482   0.5094
  38   0.8281     -0.22297   0.6852      0.17428   0.5239
  39   0.8594     -0.12611   0.6495      0.15706   0.5319
```

40	0.8850	-0.04764	0.6192	0.14656	0.5367
41	0.9000	0.01426	0.5941	0.13892	0.5402
42	0.9150	0.07503	0.5685	0.12839	0.5450
43	0.9300	0.14905	0.5356	0.11280	0.5519
44	0.9450	0.25195	0.4861	0.07598	0.5681
45	0.9600	0.29799	0.4623	0.12069	0.5484
46	0.9750	0.30517	0.4585	0.23687	0.4937
47	0.9900	0.36710	0.4240	0.33496	0.4423
48	1.0000	0.45715	0.3683	0.44929	0.3735

Appendices C Calculation of Properties for Cruising Airliner

In this section the fluid properties are calculated for typical cruise conditions. The Reynolds number is calculated for a range of Mach numbers at similar atmospheric conditions.

C.1 Chord

A typical airliner wing has a taper ratio less than unity. The chord therefore varies along the span. Figure C.1 shows the aileron from an Airbus A340, where the inboard and outboard aileron chords are 542.35 and 393.36 mm respectively to the aileron hinge line, which is at 76% of the total chord. The average chord along the aileron span can then be given by

$$c = \left(\frac{542.35 + 393.36}{2}\right)\left(\frac{1}{1 - 0.76}\right) \approx 1.95m$$

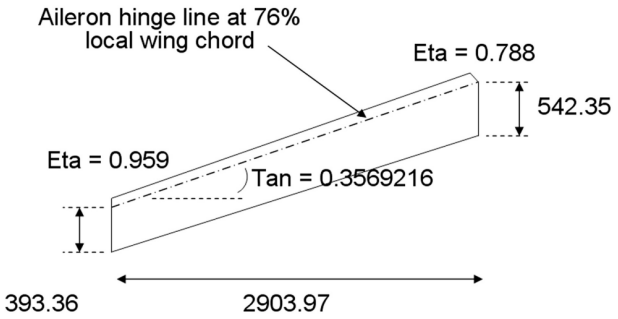


Figure C.1: Airbus datasheet excerpt showing aileron dimensions [72]

C.2 Altitude

The cruising altitude of a passenger airline is in the range of 6 000 m to more than 15000 m and can vary depending on aircraft performance, flight path, weather conditions, and other traffic. For the purpose of this dissertation a nominal value of 10 000 m will be used.

C.3 Temperature, Pressure and Density

A table of standard atmosphere values for temperature, density and pressure appears in Appendix D of [2]. These are listed in Table C.1 for the height of 10 000 m.

Table C.1: Standard atmosphere values at 10 000 m

Property	Altitude	Temperature	Pressure	Density
Value	10 000 m	223.26 K	26500 kPa	0.41351 kg/m^3

The gauge pressure is given by

$$\begin{aligned} p_{\text{gauge}} &= p_{\text{abs}} - p_{\text{atmospheric}} \\ &= 26500 - 101325 \\ &= -74825Pa \end{aligned} \tag{C.1}$$

C.4 Viscosity

Sutherland's formula for dynamic viscosity [24] gives

$$\mu = \mu_0 \left(\frac{T}{T_0}\right)^{\frac{3}{2}} \frac{T_0 + S}{T + S}$$

Here μ_0 is $1.716 \times 10^{-5} kg/ms$, T_0 is $273.16K$, and S is Sutherland's constant equal to 110.4.

This gives

$$\mu = 1.716 \times 10^{-5} \times \left(\frac{223.3}{273.15}\right)^{\frac{3}{2}} \times \frac{273.15 + 110.4}{223.3 + 110.4} \approx 1.457 \times 10^{-5} kg/ms$$

C.5 Speed of Sound

The speed of sound in air can be calculated from the formula

$$a = \sqrt{kRT}$$

where $k_{air} = 1.4$ and $R = 287$. This gives

$$a = \sqrt{1.4 \times 287 \times 223.26} \approx 300 m/s$$

C.6 Velocity

Passenger airliners typically cruise at around $M = 0.8 \pm 0.02$, where

$$M = \frac{V}{a}$$

Taking the nominal value of $M = 0.8$, the velocity becomes

$$V = 0.8 \times 300 = 240\ m/s$$

C.7 Reynolds Number

The Reynolds number is given by

$$\begin{aligned} Re &= \frac{\rho V c}{\mu} \\ &= \frac{\rho M a\, c}{\mu} \end{aligned}$$

The Reynolds number for flight at typical cruise conditions then becomes

$$Re = \frac{0.411 \times 0.8 \times 300 \times 1.95}{1.457 \times 10^{-5}} \approx 1.3 \times 10^{7} \tag{C.2}$$

Table C.2 shows the Reynolds numbers used during the simulations of Chapter 4. These were obtained using the values as in equation C.2 and varying only the value of M.

Table C.2: Reynolds number for range of Mach numbers at atmospheric conditions typical of airliner cruise altitudes

Mach Number	Reynolds Number
0.1	1.656×10^{6}
0.2	3.313×10^{6}
0.3	4.969×10^{6}
0.4	6.626×10^{6}
0.5	8.282×10^{6}
0.6	9.938×10^{6}
0.7	1.159×10^{7}
0.8	1.325×10^{7}
0.9	1.491×10^{7}

Appendices D CFD Output

D.1 Lift and Drag History Plots

Contained in this appendix are the plots of C_L and C_D for each iteration of the CFD solver. These illustrate the ease or difficulty encountered in obtaining the final converged result by the total number of iterations undertaken as well as the change in value of the parameter recorded.

The plots begin overleaf. A summary of all the coefficients calculated appear at the end of the section on page D-22.

D.1.1 α-sweep at $M = 0.5$

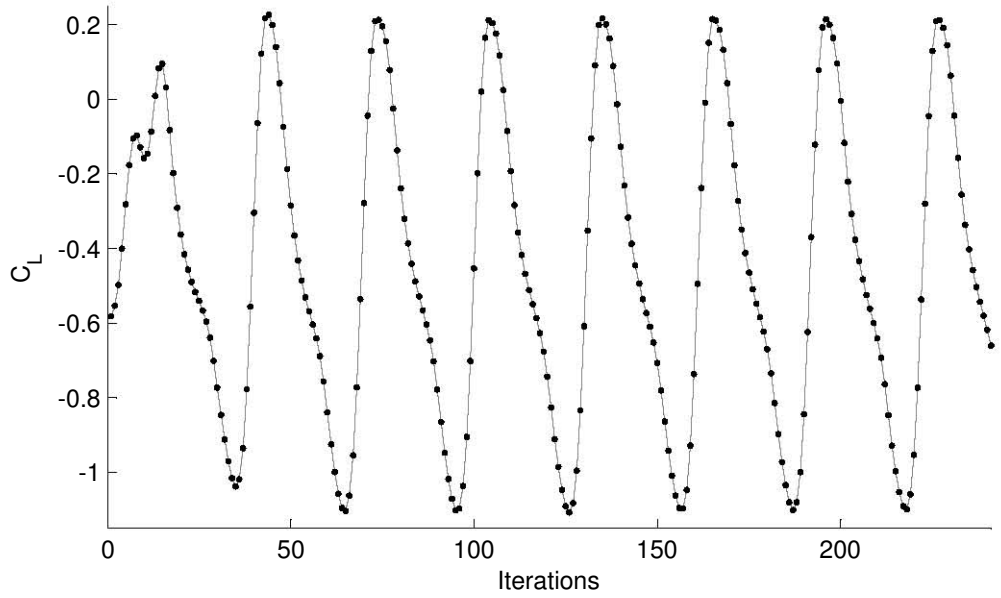


Figure D.1: Lift coefficient history for $\alpha = -9°$, $M_\infty = 0.5$

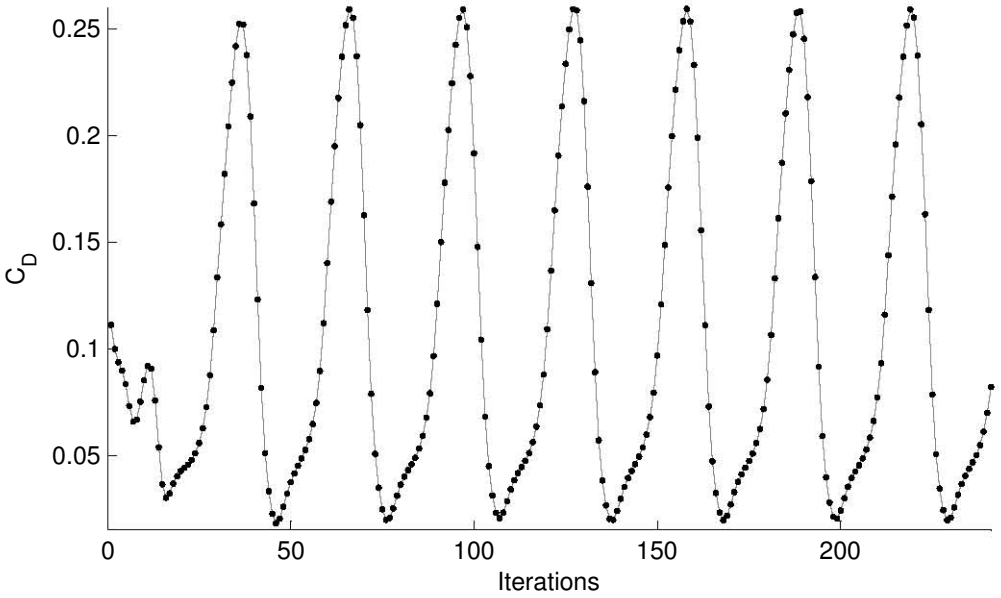


Figure D.2: Drag coefficient history for $\alpha = -9°$, $M_\infty = 0.5$

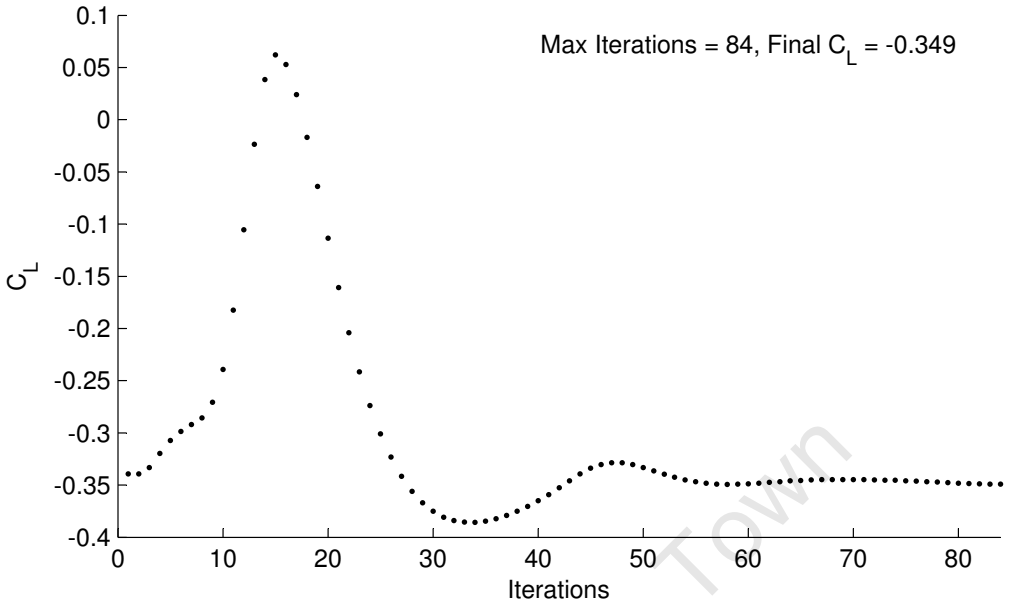


Figure D.3: Lift coefficient history for $\alpha = -6°$, $M_\infty = 0.5$

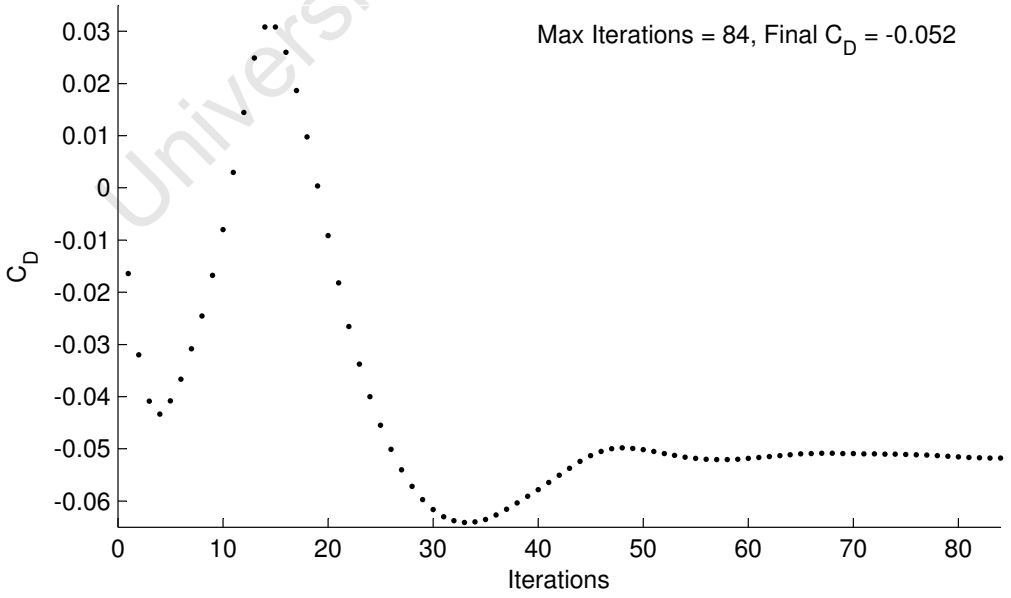


Figure D.4: Drag coefficient history for $\alpha = -6°$, $M_\infty = 0.5$

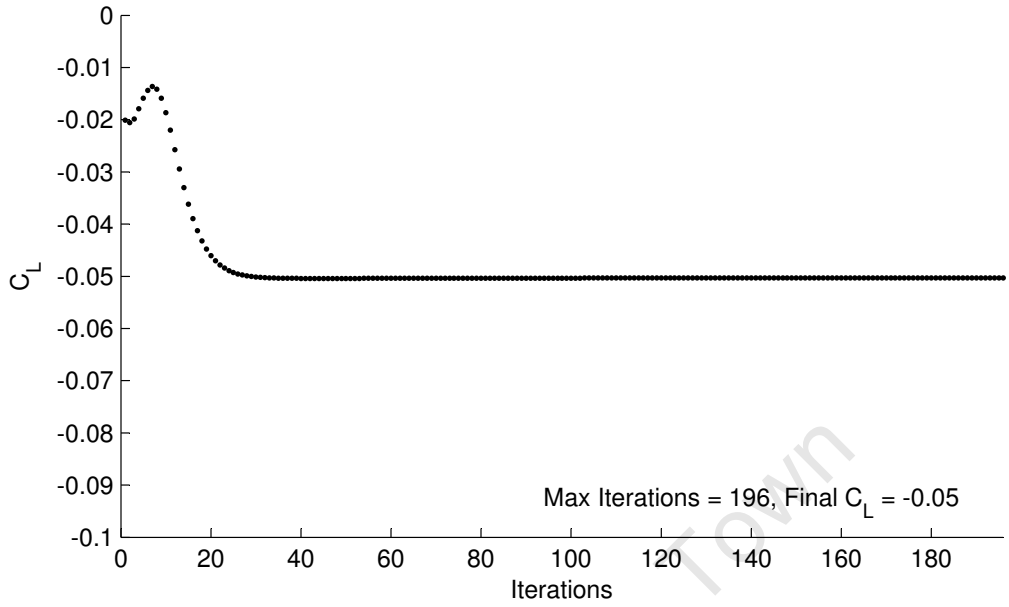


Figure D.5: Lift coefficient history for $\alpha = -3°$, $M_\infty = 0.5$

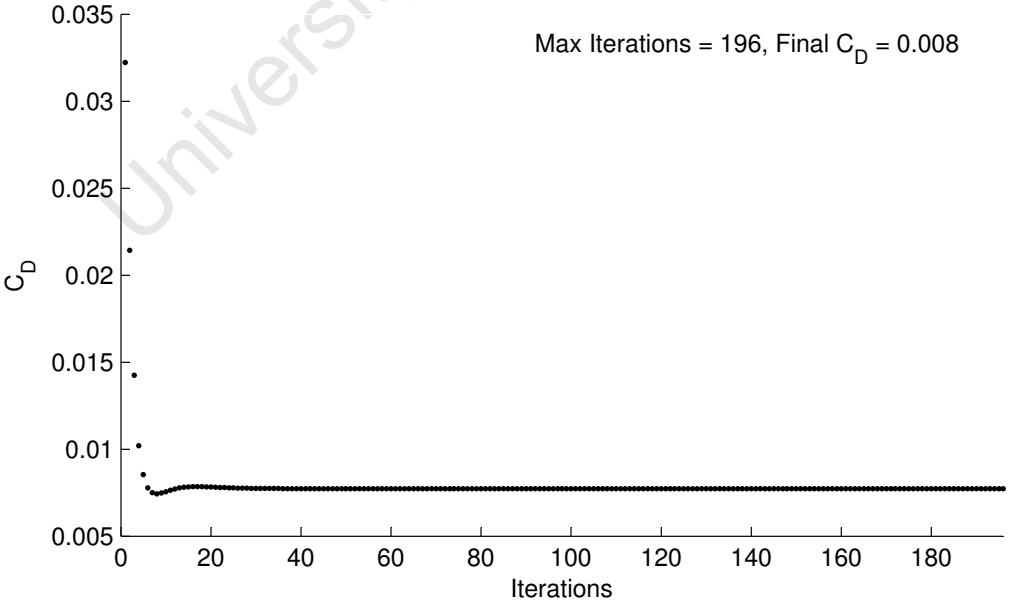


Figure D.6: Drag coefficient history for $\alpha = -3°$, $M_\infty = 0.5$

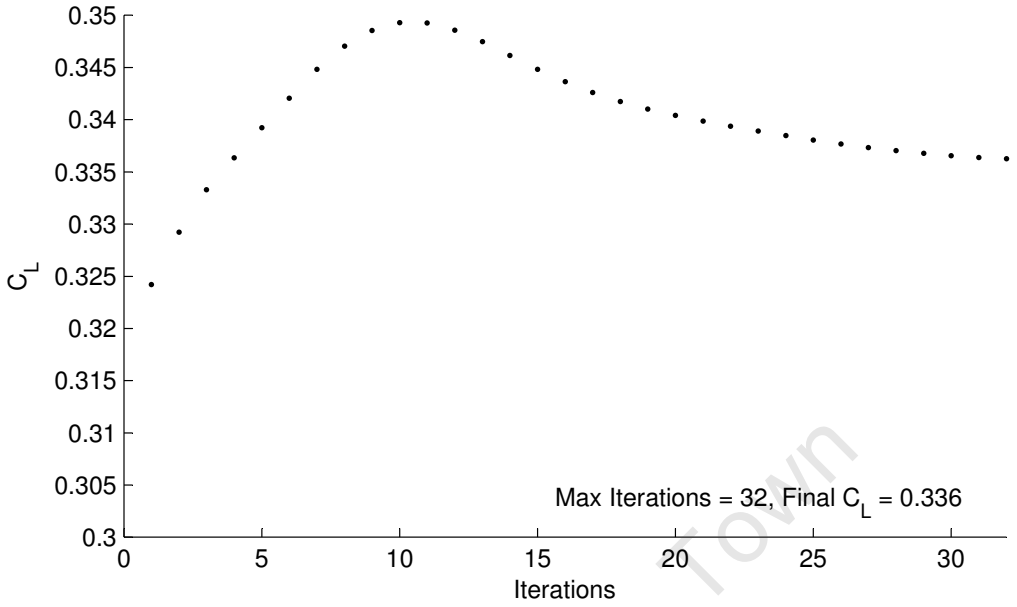


Figure D.7: Lift coefficient history for $\alpha = 0°$, $M_\infty = 0.5$

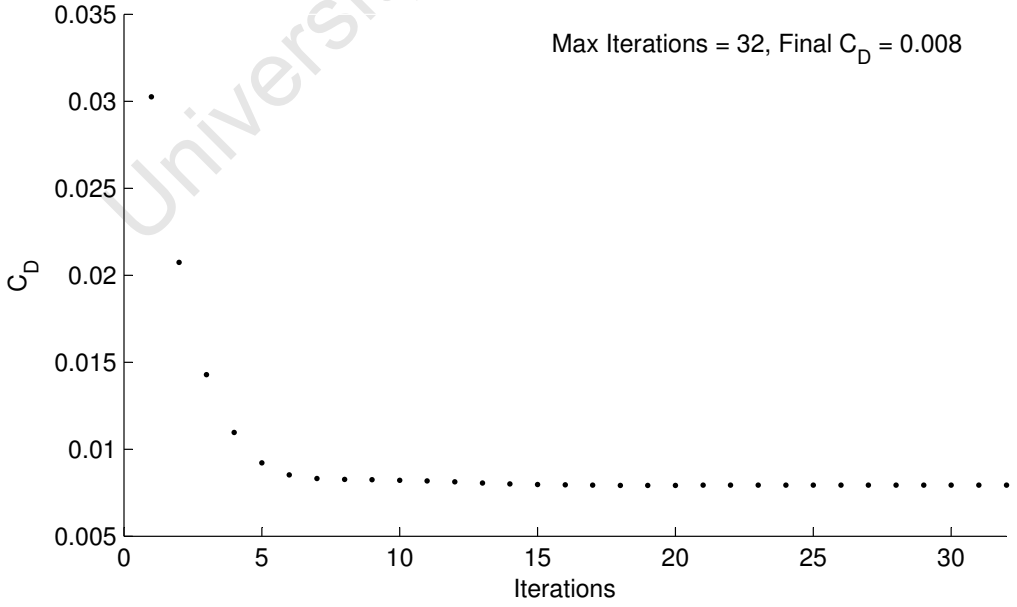


Figure D.8: Drag coefficient history for $\alpha = 0°$, $M_\infty = 0.5$

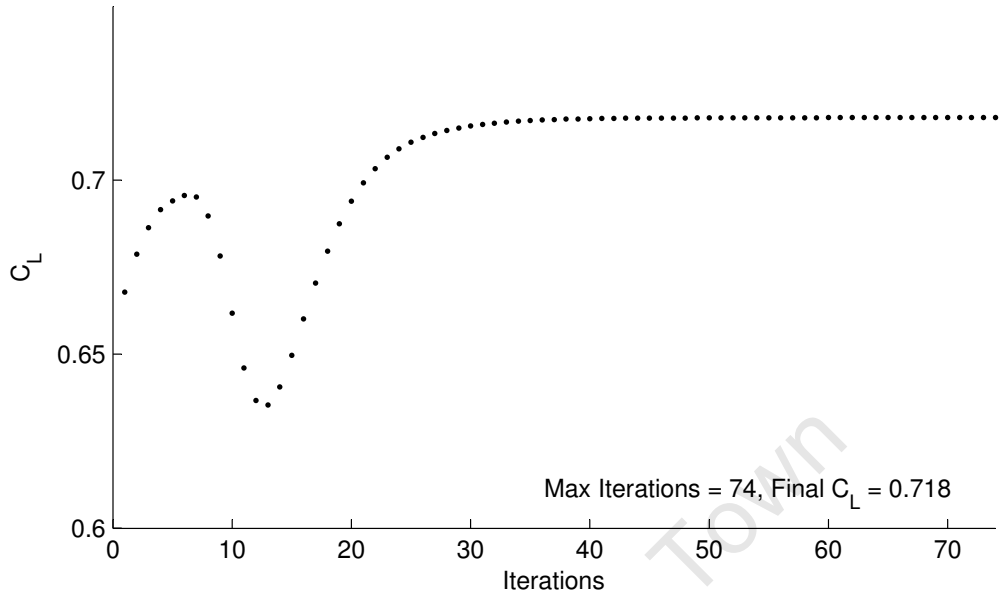


Figure D.9: Lift coefficient history for $\alpha = 3°$, $M_\infty = 0.5$

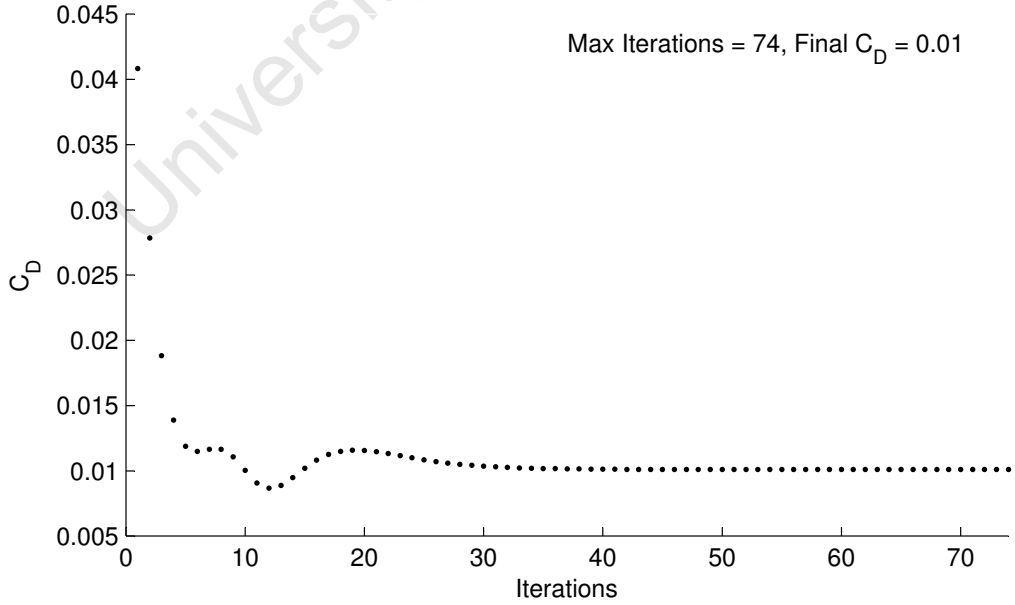


Figure D.10: Drag coefficient history for $\alpha = 3°$, $M_\infty = 0.5$

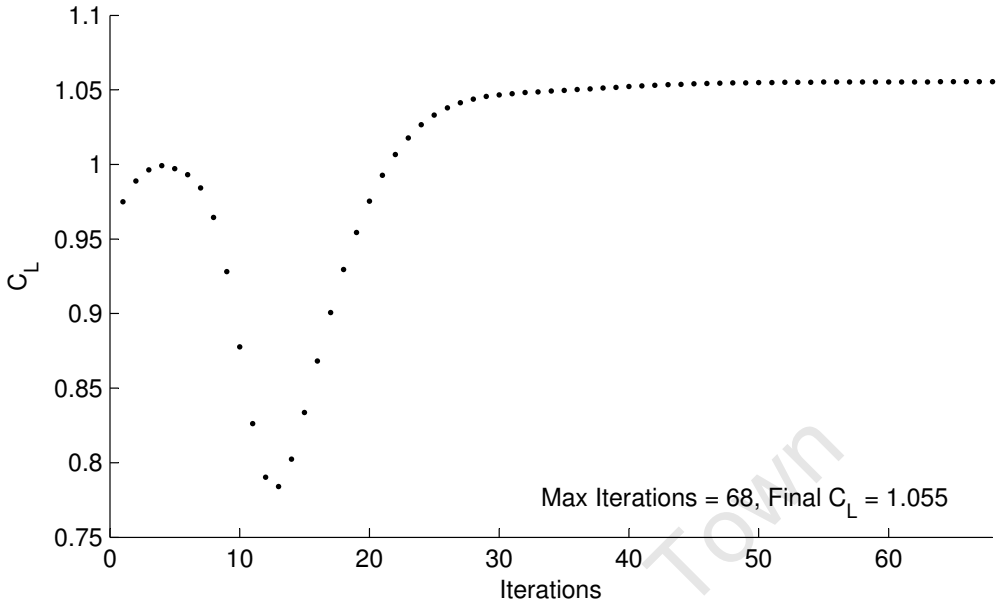


Figure D.11: Lift coefficient history for $\alpha = 6°$, $M_\infty = 0.5$

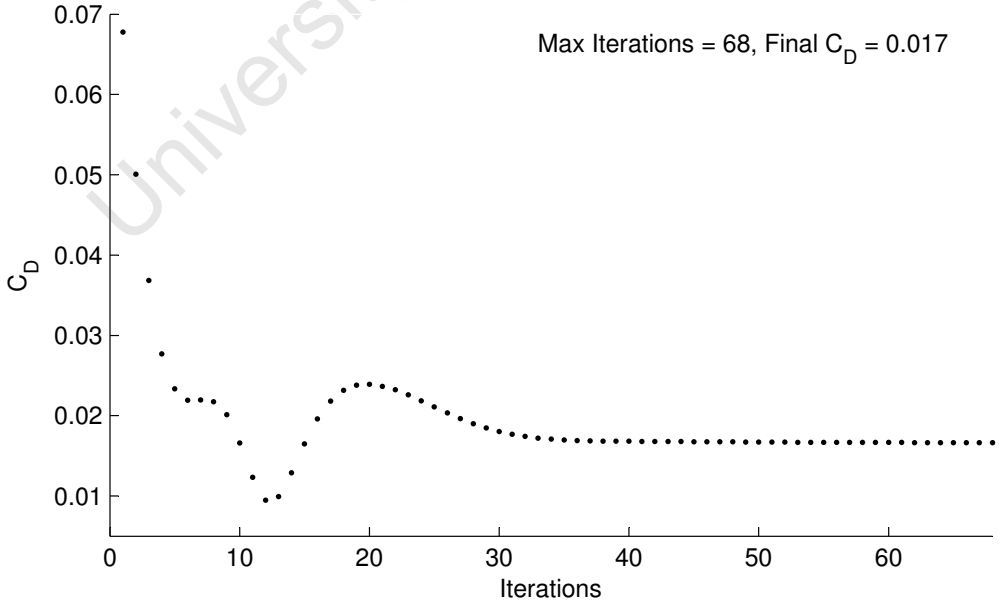


Figure D.12: Drag coefficient history for $\alpha = 6°$, $M_\infty = 0.5$

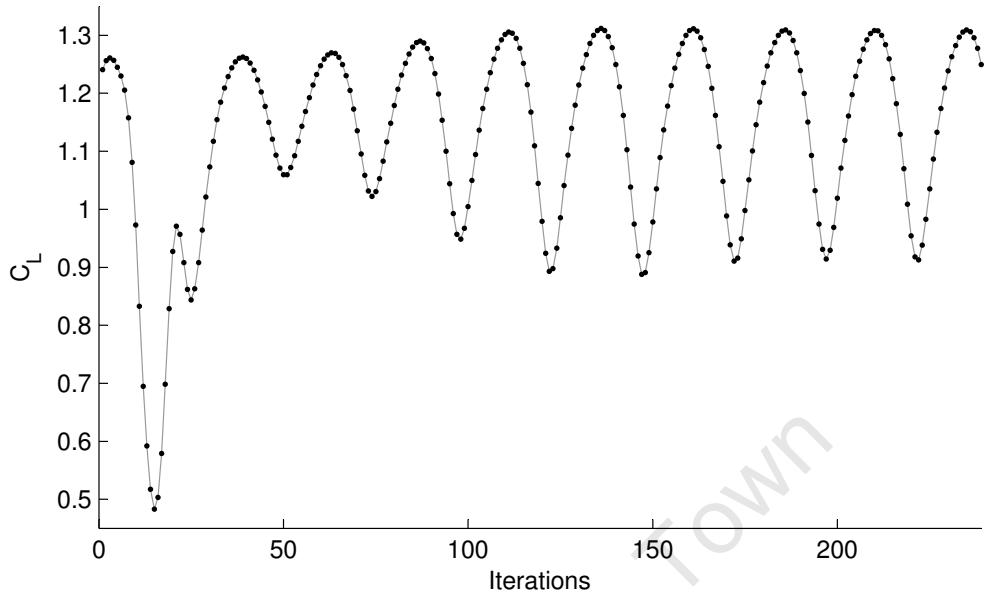


Figure D.13: Lift coefficient history for $\alpha = 9°$, $M_\infty = 0.5$

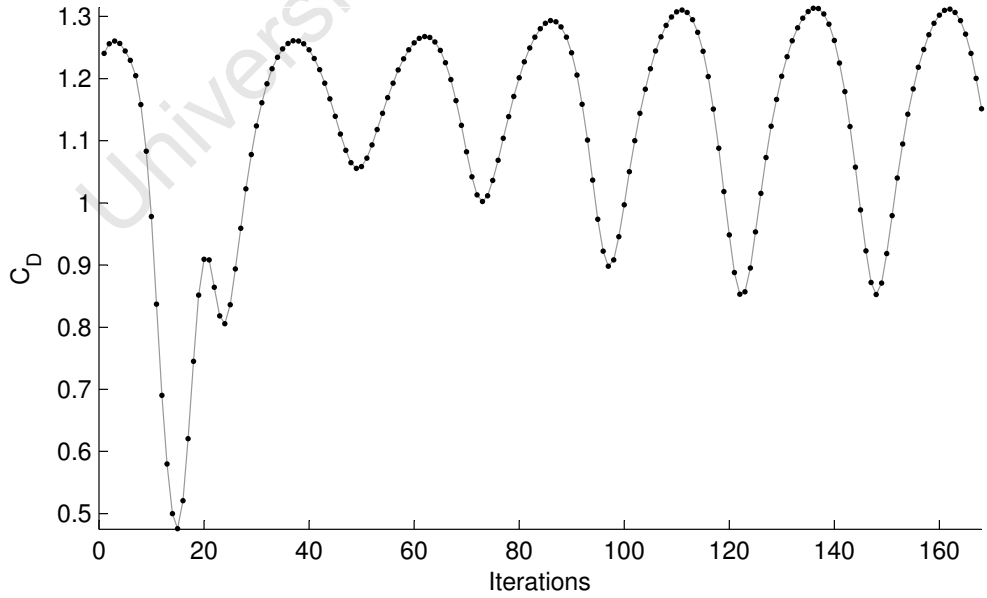


Figure D.14: Drag coefficient history for $\alpha = 9°$, $M_\infty = 0.5$

D.1.2 α-sweep at $M = 0.8$

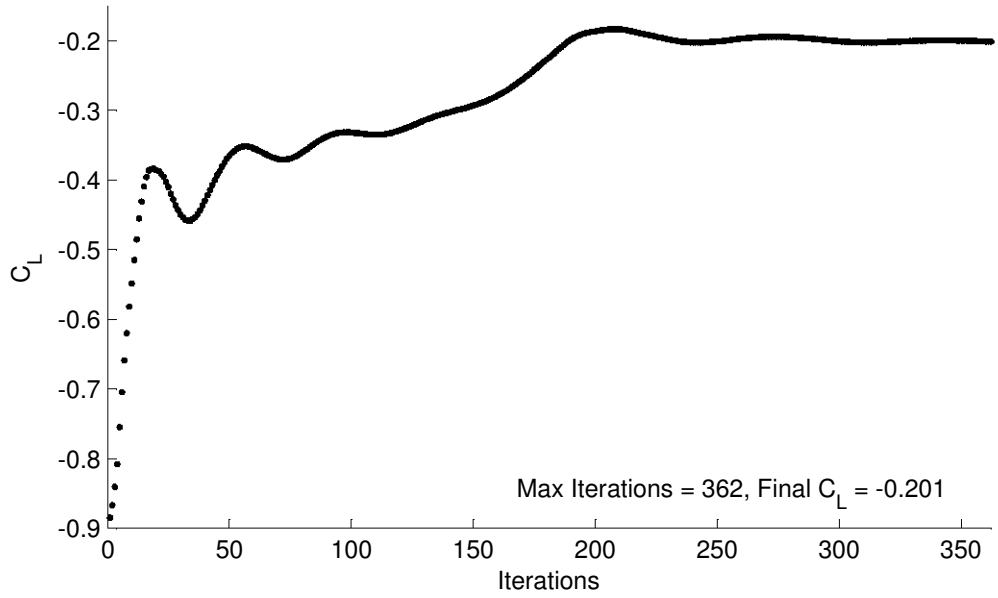


Figure D.15: Lift coefficient history for $\alpha = -9°$, $M_\infty = 0.8$

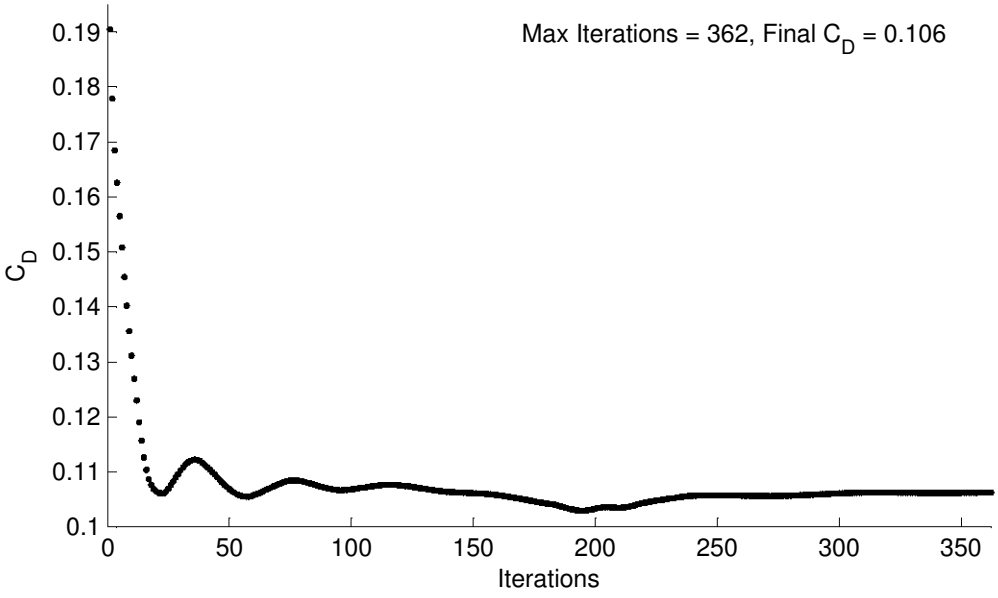


Figure D.16: Drag coefficient history for $\alpha = -9°$, $M_\infty = 0.8$

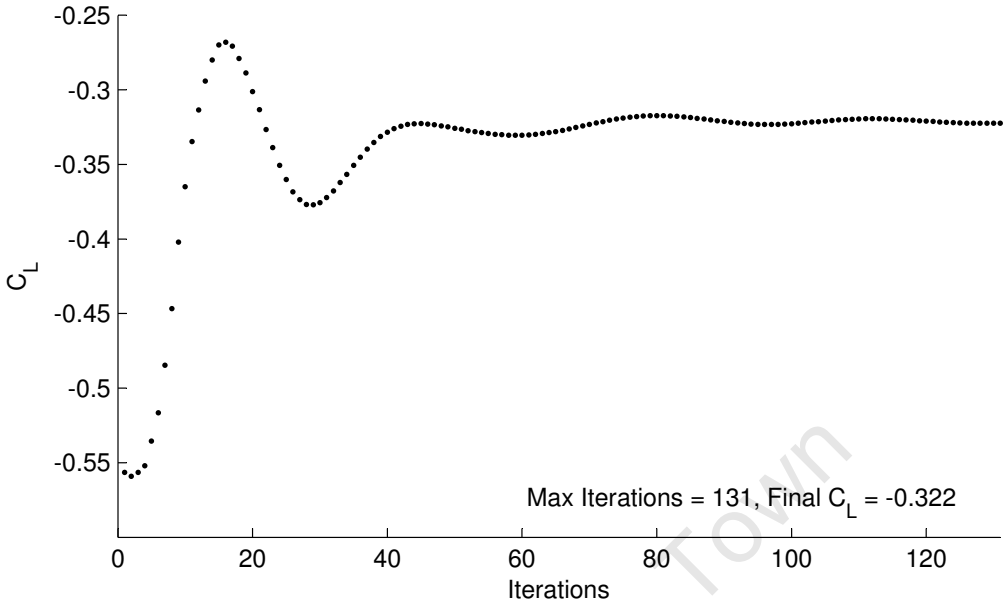


Figure D.17: Lift coefficient history for $\alpha = -6°$, $M_\infty = 0.8$

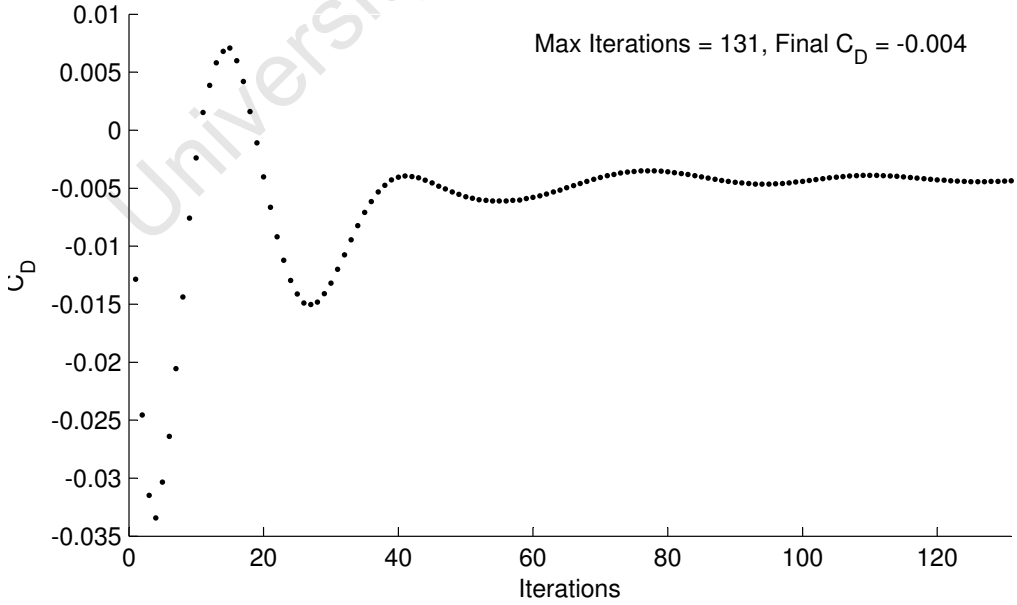


Figure D.18: Drag coefficient history for $\alpha = -6°$, $M_\infty = 0.8$

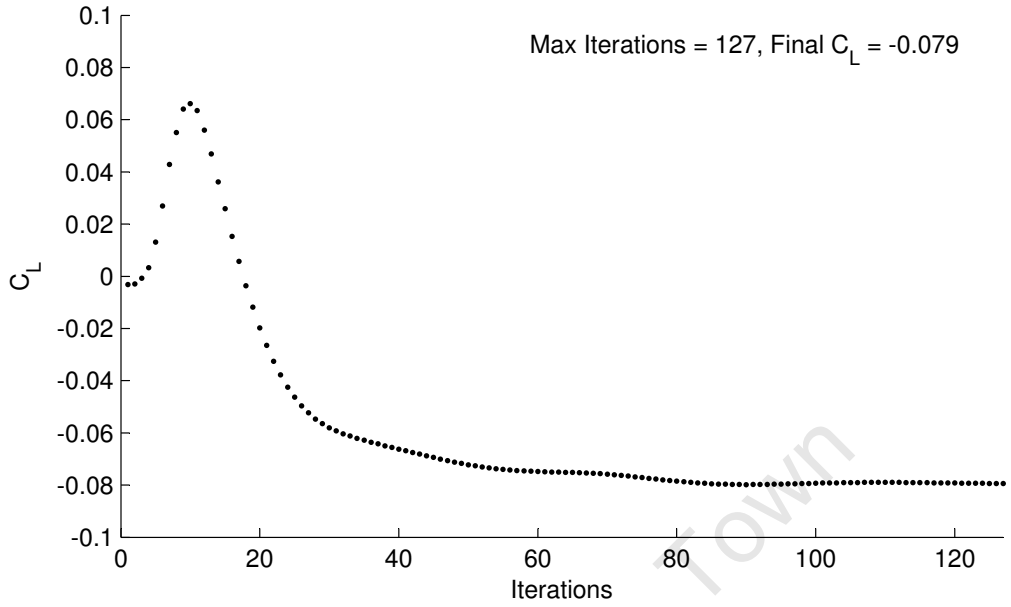


Figure D.19: Lift coefficient history for $\alpha = -3°$, $M_\infty = 0.8$

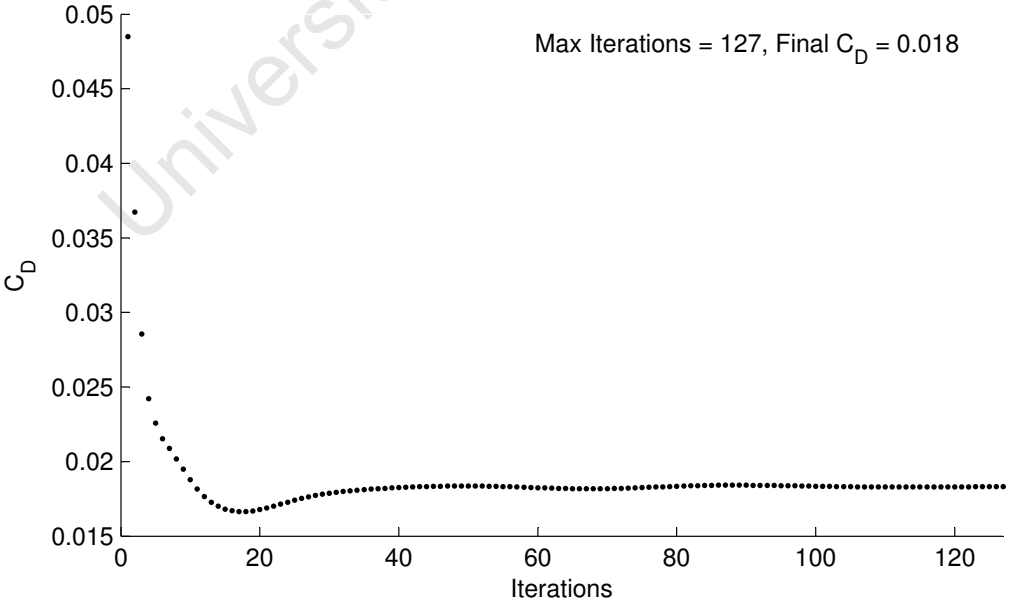


Figure D.20: Drag coefficient history for $\alpha = -3°$, $M_\infty = 0.8$

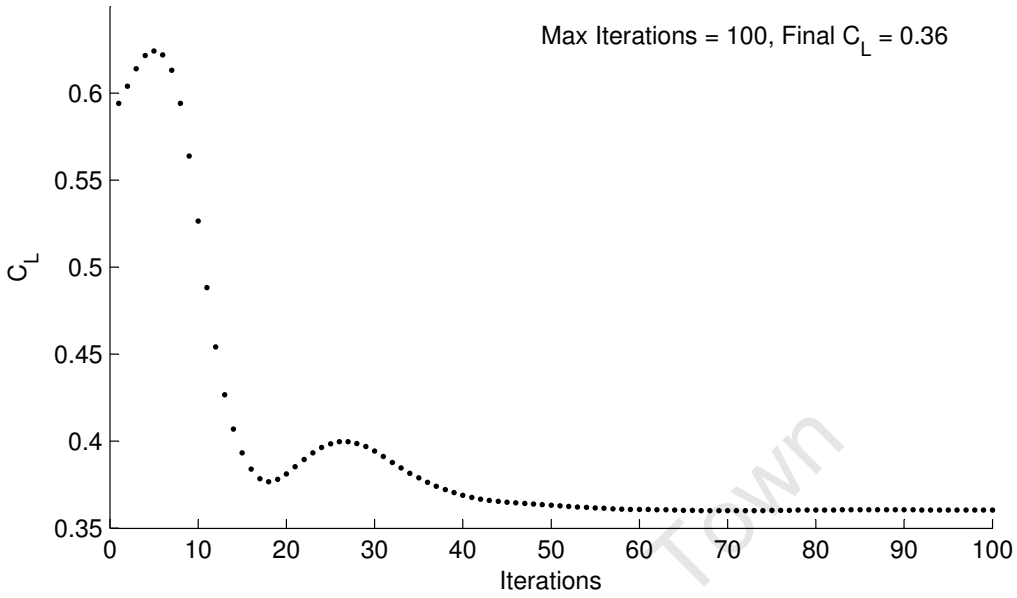


Figure D.21: Lift coefficient history for $\alpha = 0°$, $M_\infty = 0.8$

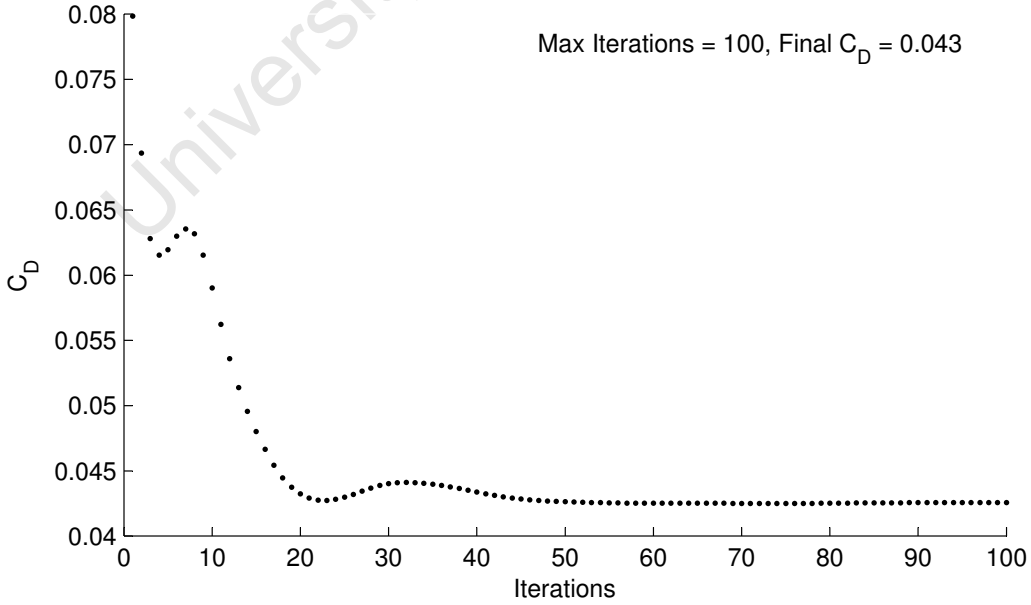


Figure D.22: Drag coefficient history for $\alpha = 0°$, $M_\infty = 0.8$

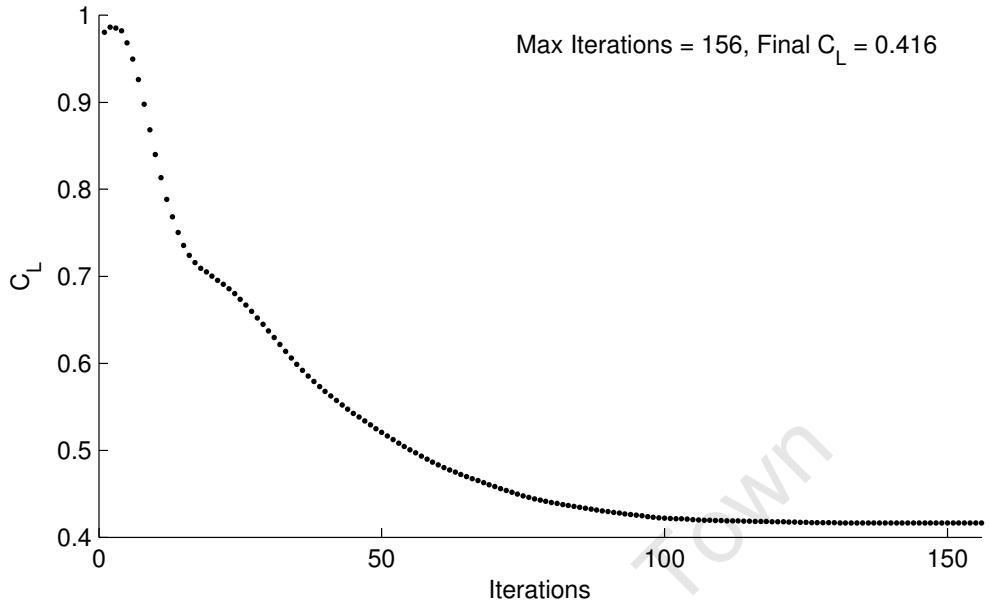


Figure D.23: Lift coefficient history for $\alpha = 3°$, $M_\infty = 0.8$

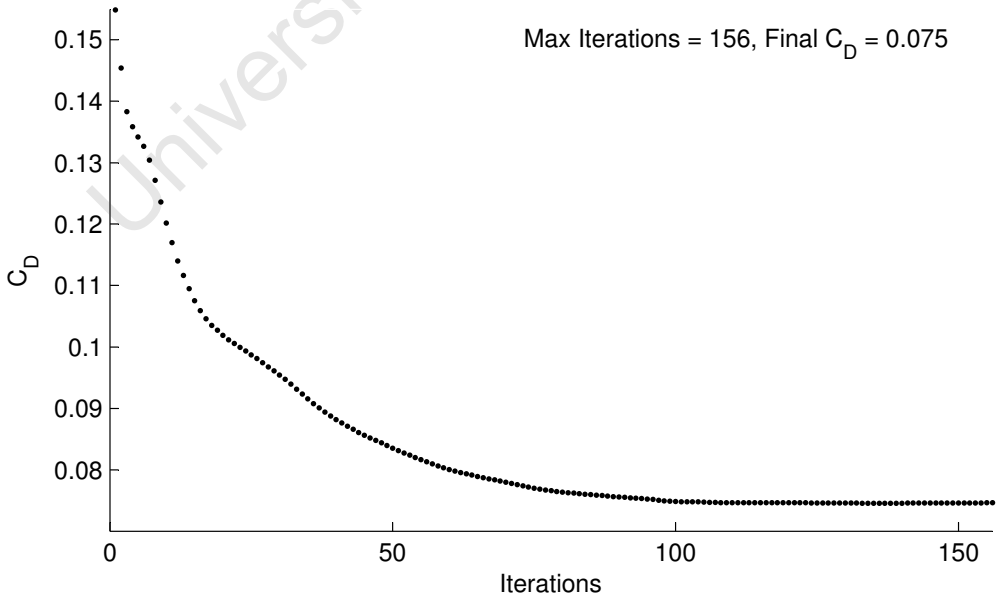


Figure D.24: Drag coefficient history for $\alpha = 3°$, $M_\infty = 0.8$

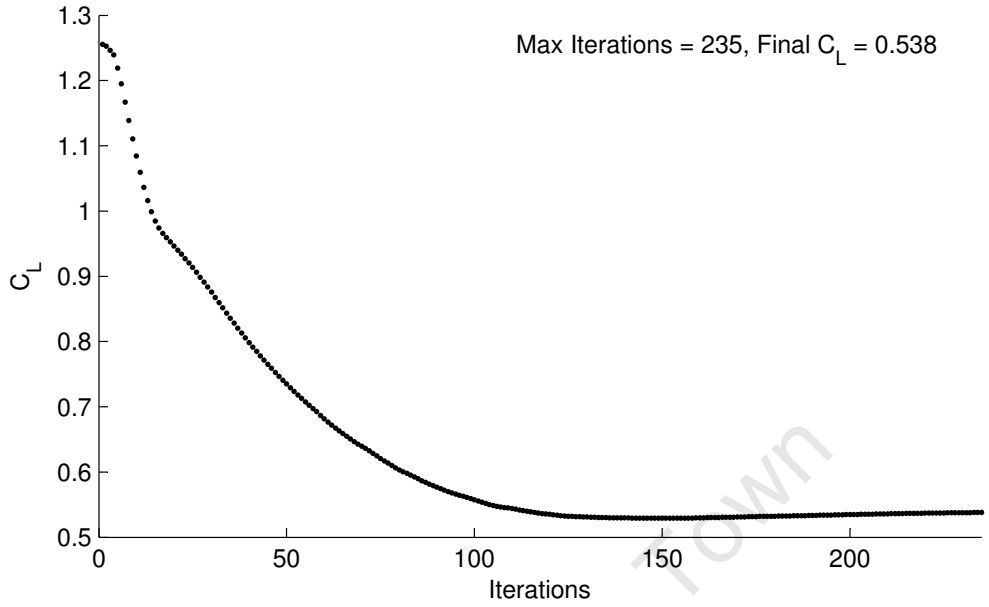


Figure D.25: Lift coefficient history for $\alpha = 6°$, $M_\infty = 0.8$

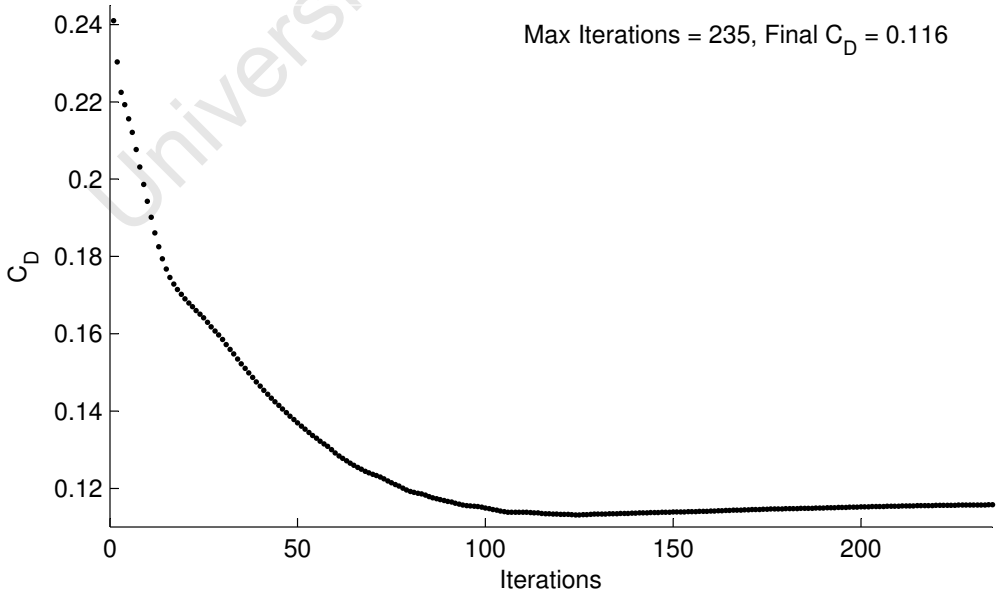


Figure D.26: Drag coefficient history for $\alpha = 6°$, $M_\infty = 0.8$

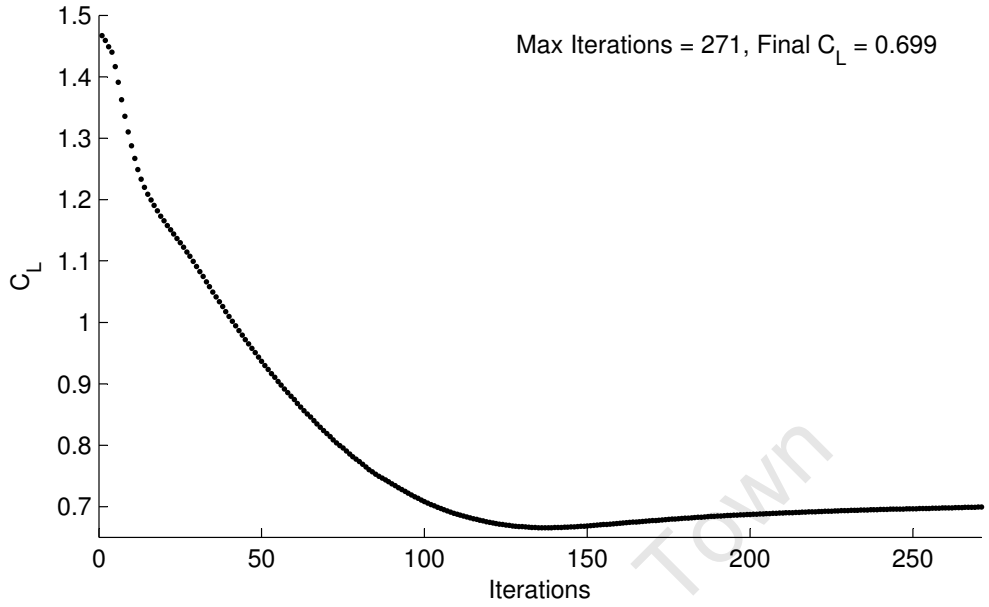


Figure D.27: Lift coefficient history for $\alpha = 9°$, $M_\infty = 0.8$

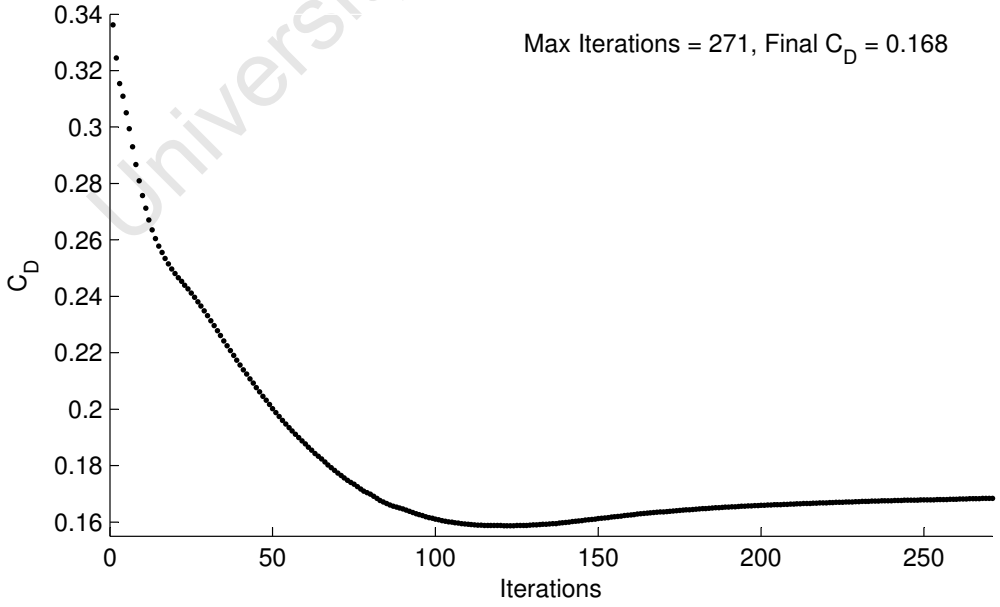


Figure D.28: Drag coefficient history for $\alpha = 9°$, $M_\infty = 0.8$

D.1.3 Mach-sweep at $\alpha = 3°$

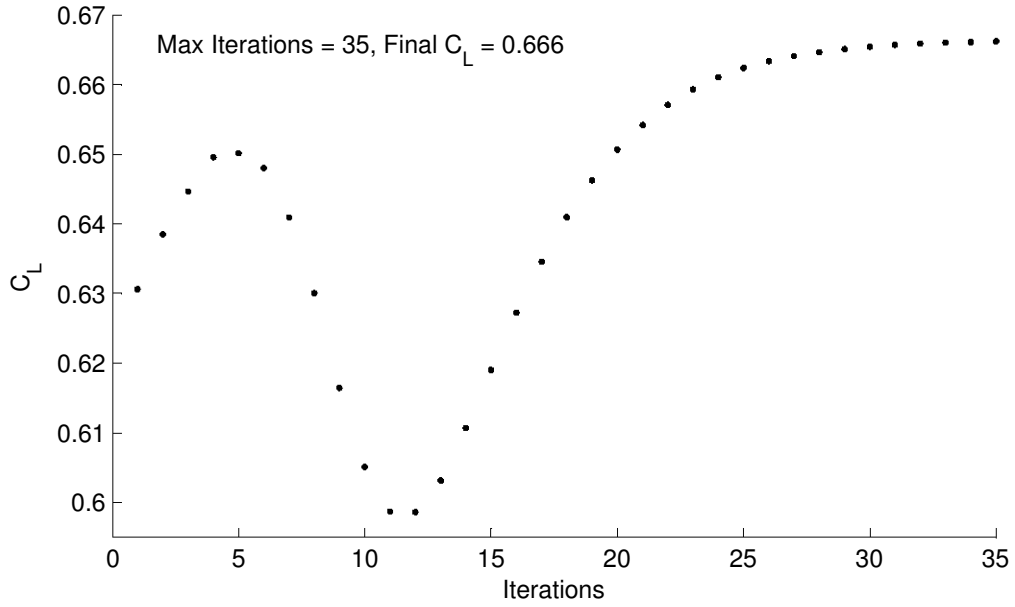


Figure D.29: Lift coefficient history for $M_\infty = 0.4$, $\alpha = 3°$

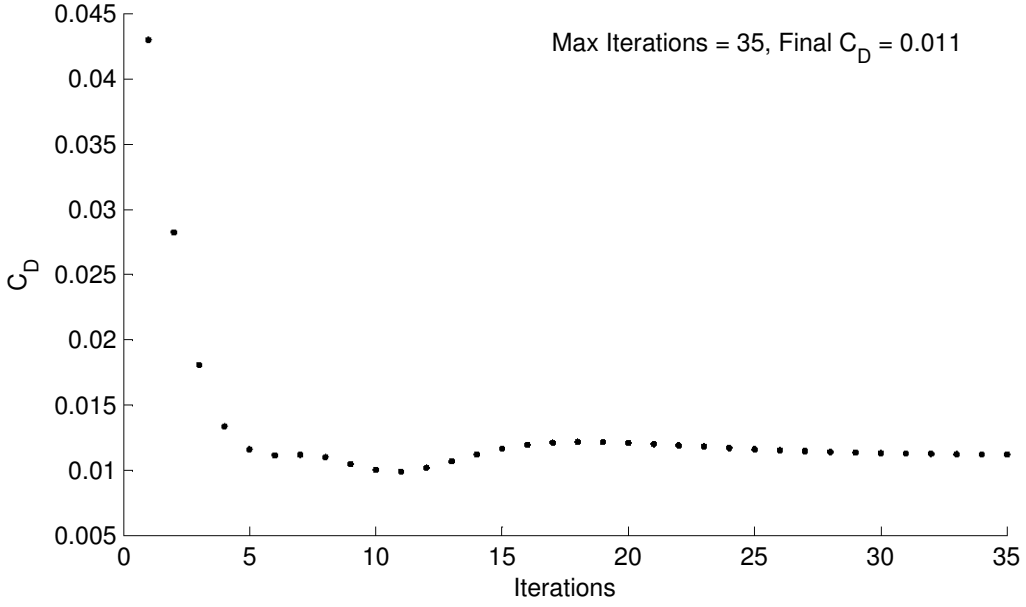


Figure D.30: Drag coefficient history for $M_\infty = 0.4$, $\alpha = 3°$

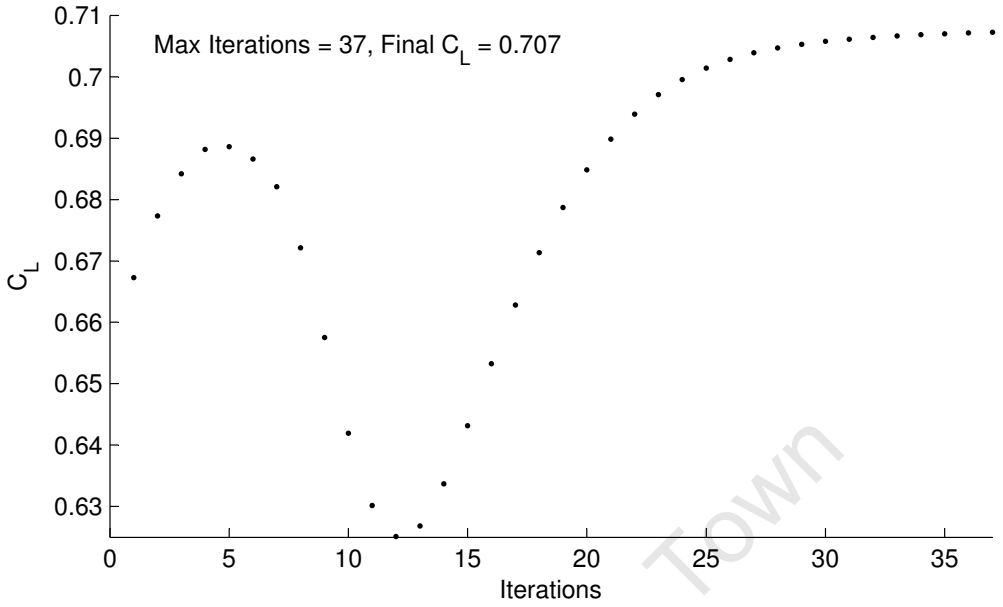


Figure D.31: Lift coefficient history for $M_\infty = 0.5$, $\alpha = 3°$

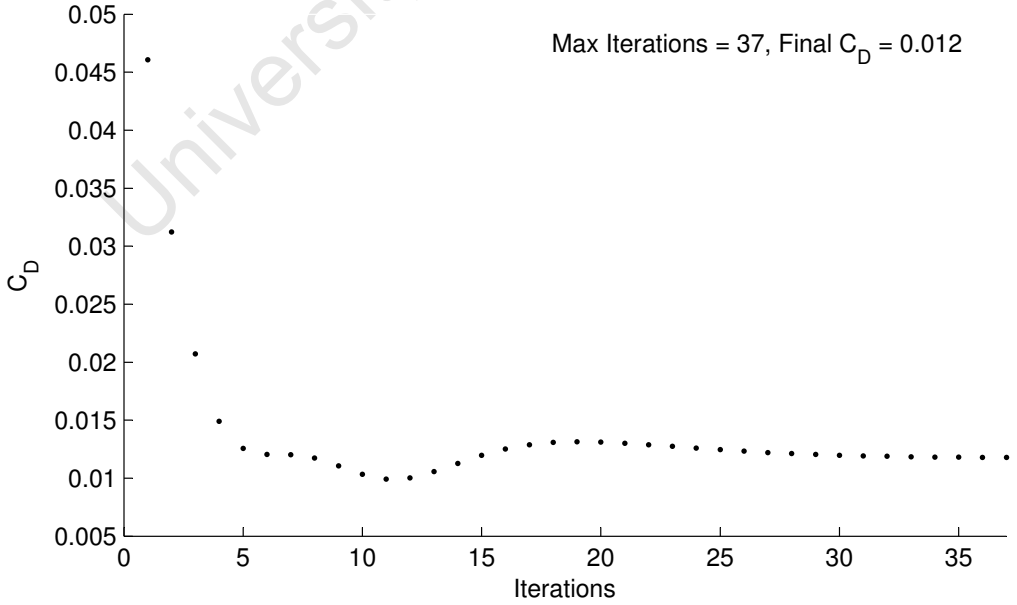


Figure D.32: Drag coefficient history for $M_\infty = 0.5$, $\alpha = 3°$

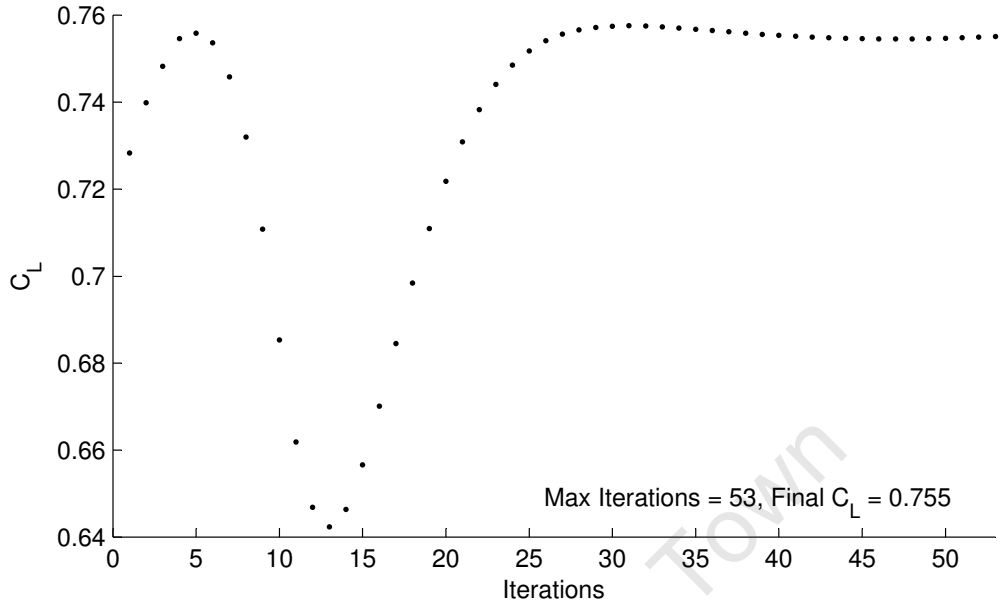


Figure D.33: Lift coefficient history for $M_\infty = 0.6$, $\alpha = 3°$

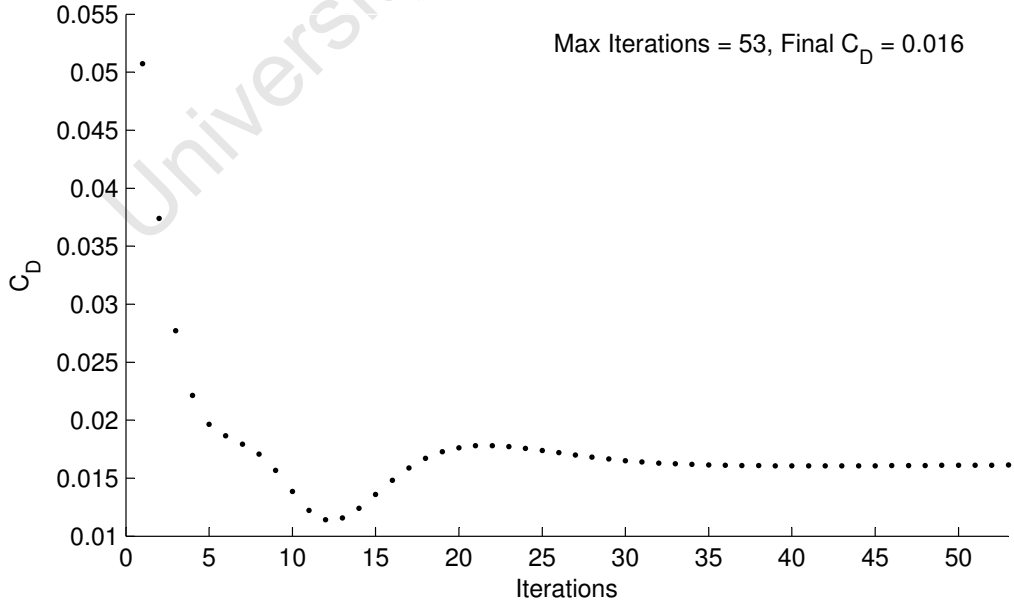


Figure D.34: Drag coefficient history for $M_\infty = 0.6$, $\alpha = 3°$

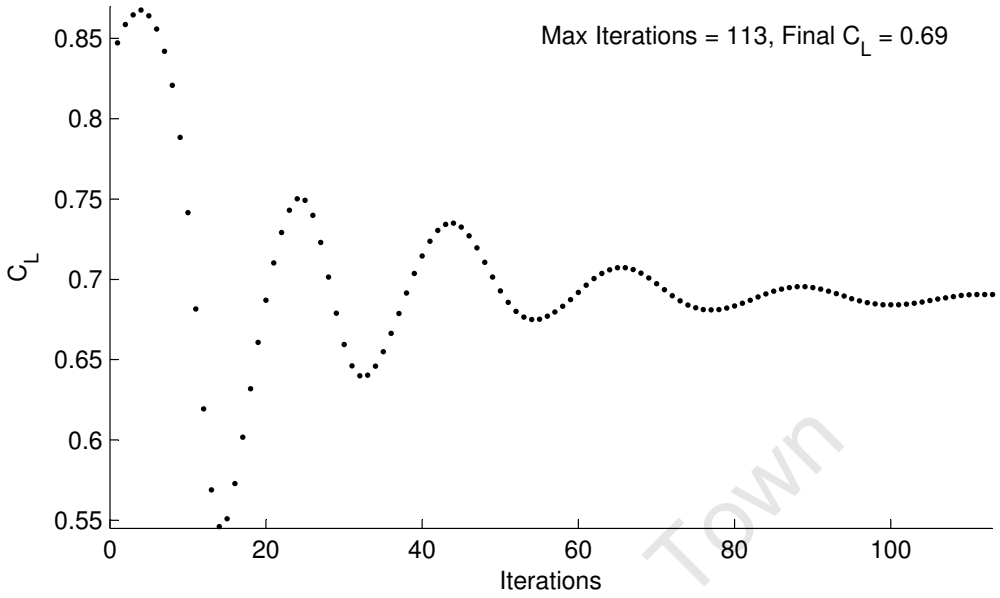


Figure D.35: Lift coefficient history for $M_\infty = 0.7$, $\alpha = 3°$

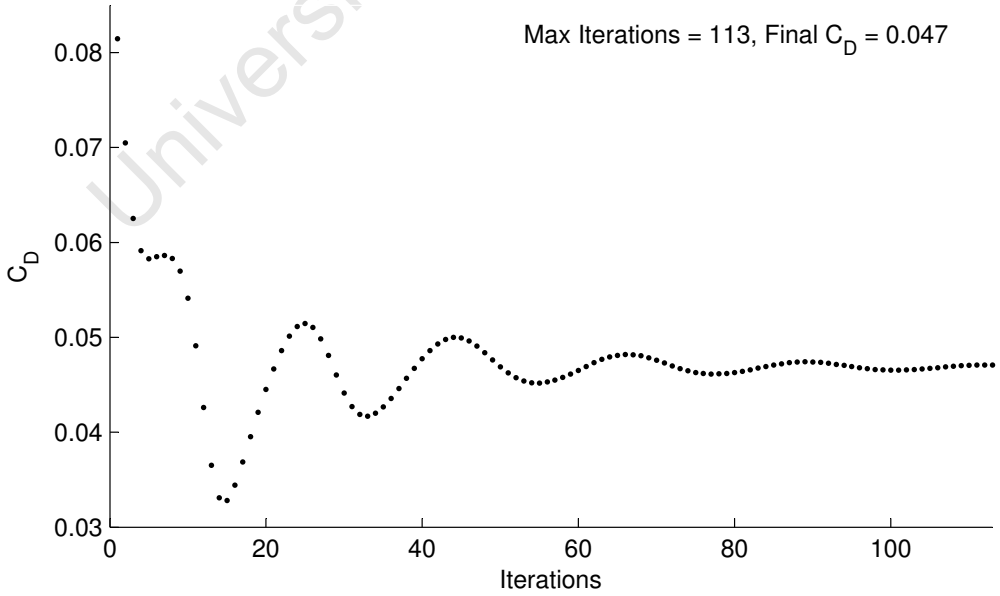


Figure D.36: Drag coefficient history for $M_\infty = 0.7$, $\alpha = 3°$

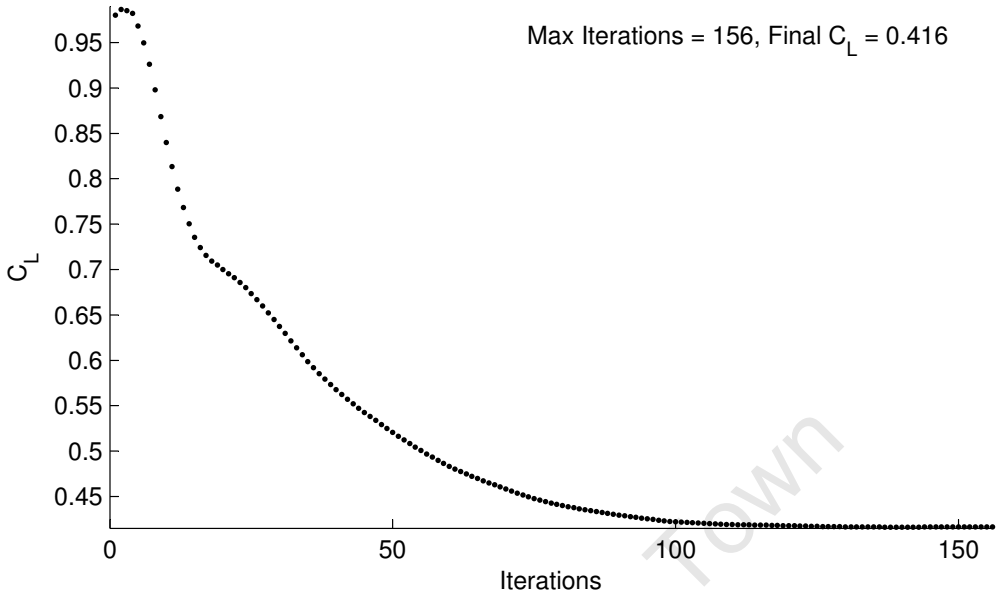


Figure D.37: Lift coefficient history for $M_\infty = 0.8$, $\alpha = 3°$

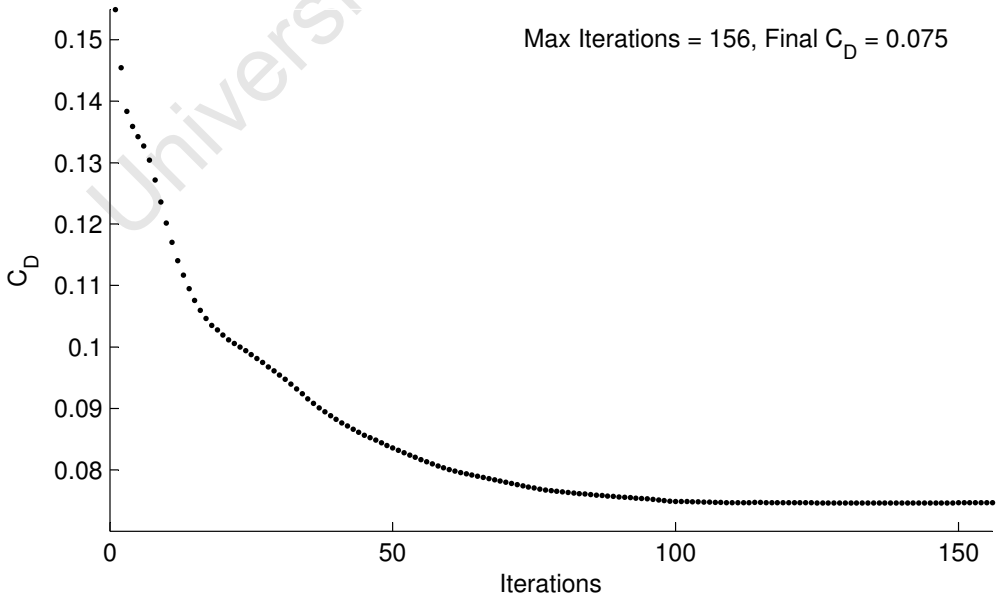


Figure D.38: Drag coefficient history for $M_\infty = 0.8$, $\alpha = 3°$

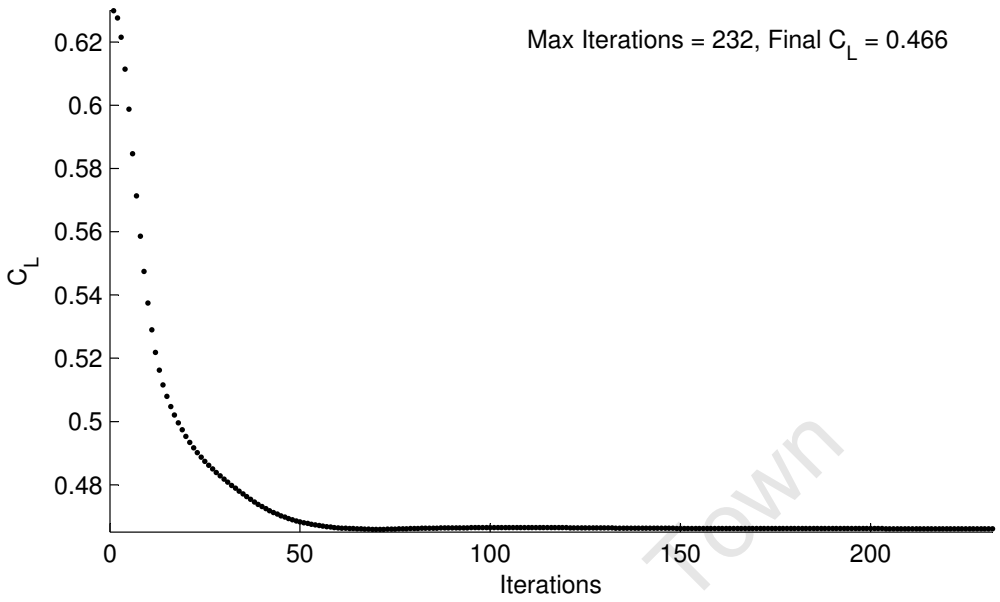


Figure D.39: Lift coefficient history for $M_\infty = 0.9$, $\alpha = 3°$

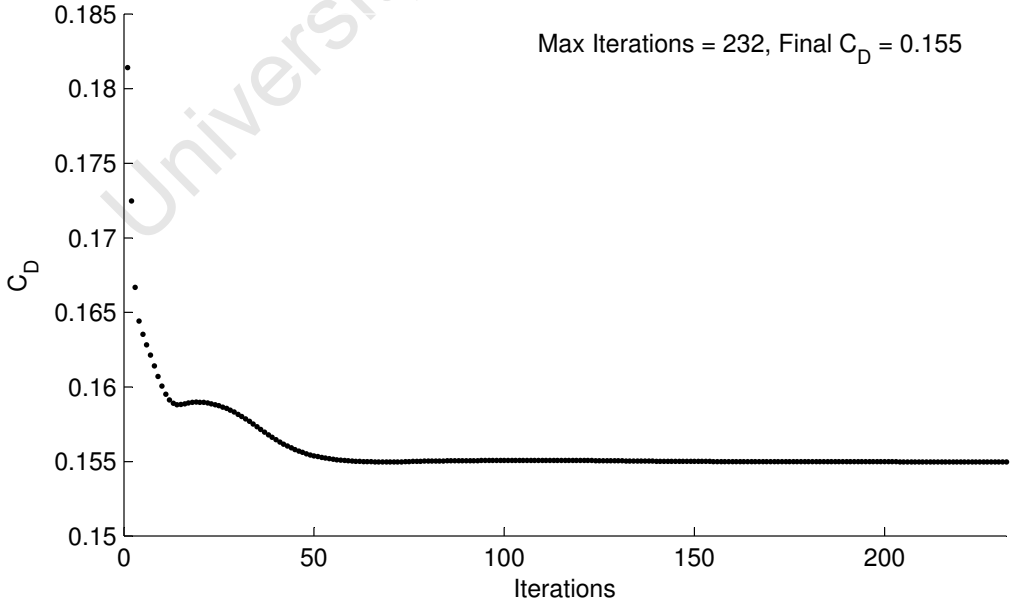


Figure D.40: Drag coefficient history for $M_\infty = 0.9$, $\alpha = 3°$

Table D.1: Final Values of C_L and C_D for α-sweep at Mach 0.5

α	C_L	C_D
-6	-0.349	-0.0518
-3	-0.050	0.0077
0	0.336	0.0079
3	0.718	0.0101
6	1.055	0.0166

Table D.2: Final Values of C_L and C_D for α-sweep at Mach 0.8

α	C_L	C_D
-9	-0.201	0.1063
-6	-0.322	-0.0044
-3	-0.079	0.0183
0	0.360	0.0426
3	0.416	0.0746
6	0.538	0.1158
9	0.699	0.1684

Table D.3: Final Values of C_L and C_D for Mach-sweep at $\alpha = 3°$

M	C_L	C_D
0.4	0.666	0.0112
0.5	0.707	0.0118
0.6	0.755	0.0161
0.7	0.690	0.0471
0.8	0.416	0.0746
0.9	0.466	0.1550

D.2 Wall $y+$ Values and Example Plots

Tables D.5 to D.4 show that for all simulations undertaken the $y+$ values were within the range required by the Spalart-Allmaras turbulence model. Examples of the full results for a selection of simulations are plotted in Figs. D.41 toD.44. Here the sample was chosen to represent the variety of plots found, so that all plots displayed similar characteristics to those presented here. Details of the simulation pertaining to each plot can be found in the respective caption.

Table D.4: Minimum and maximum values of $y+$ for α-sweep at $M = 0.5$

α	$y+_{\text{min}}$	$y+_{\text{max}}$
-6	228.88	43.62
-3	580.74	58.49
0	171.58	43.20
3	558.14	76.70
6	181.66	40.47

Table D.5: Minimum and maximum values of $y+$ for α-sweep at $M = 0.8$

α	$y+_{\text{min}}$	$y+_{\text{max}}$
-9	38.33	232.1
-6	39.21	234.7
-3	41.94	232.8
0	40.55	226.0
3	44.31	221.6
6	36.68	220.2
9	46.94	215.0

Table D.6: Minimum and maximum values of $y+$ for Mach-sweep at $\alpha = 3°$

M	$y+_{\text{min}}$	$y+_{\text{max}}$
0.4	35.49	141.48
0.5	42.67	179.15
0.6	41.05	184.62
0.7	37.91	203.50
0.8	48.96	202.40
0.9	38.50	231.13

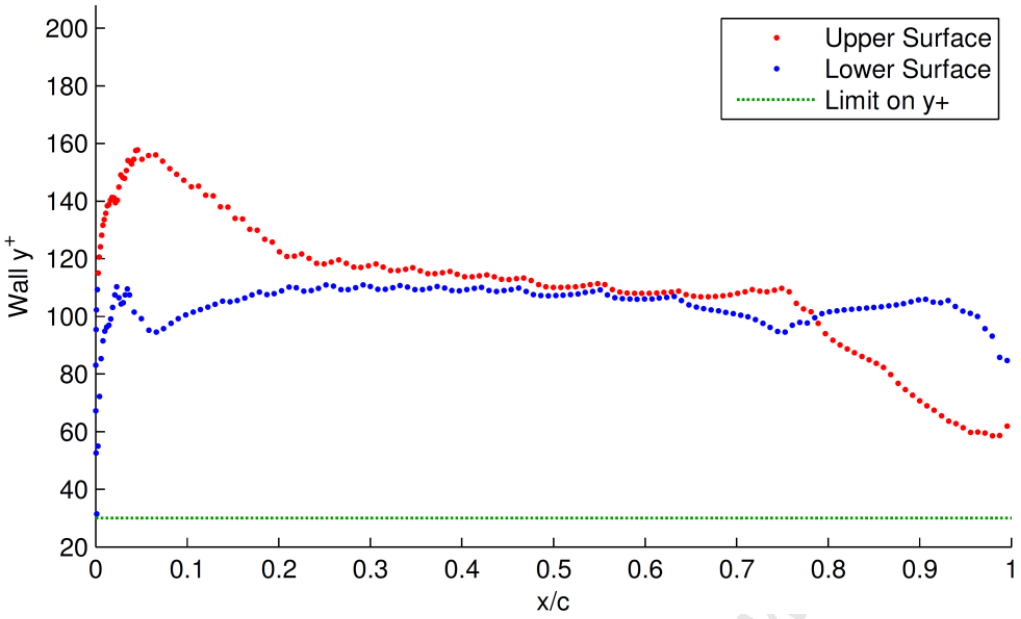


Figure D.41: Wall $y+$ for $M = 0.4$, $\alpha = 3°$

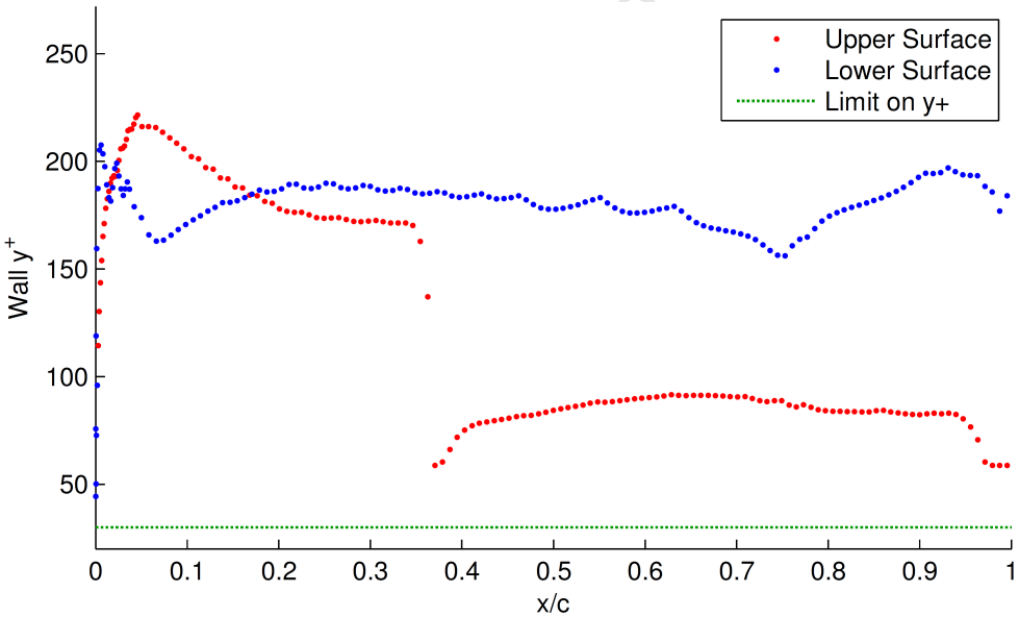


Figure D.42: Wall $y+$ for $M = 0.8$, $\alpha = 3°$

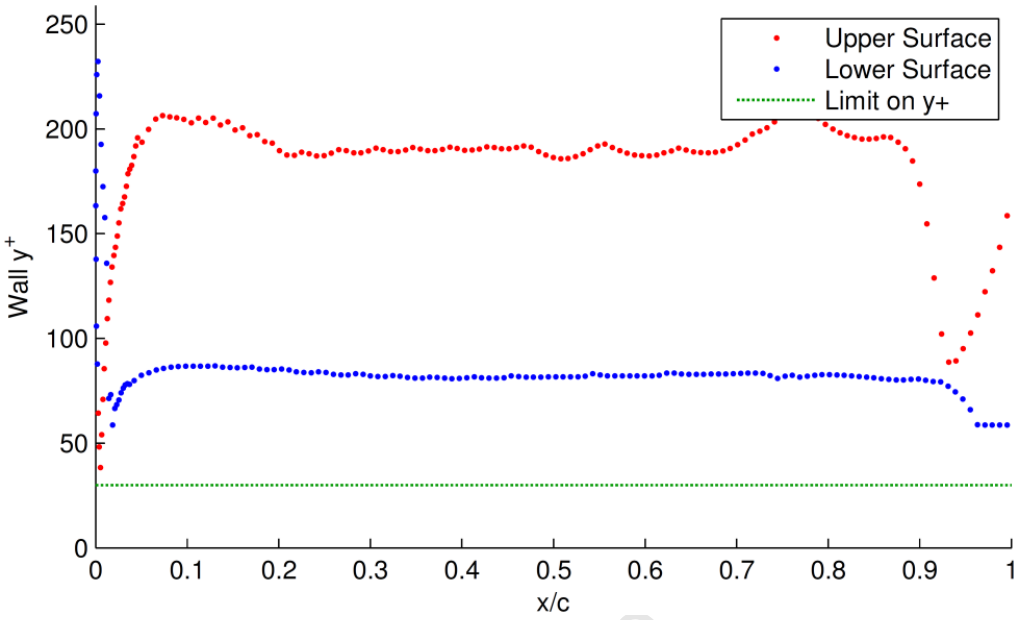


Figure D.43: Wall $y+$ for $\alpha = -9°$, $M = 0.8$

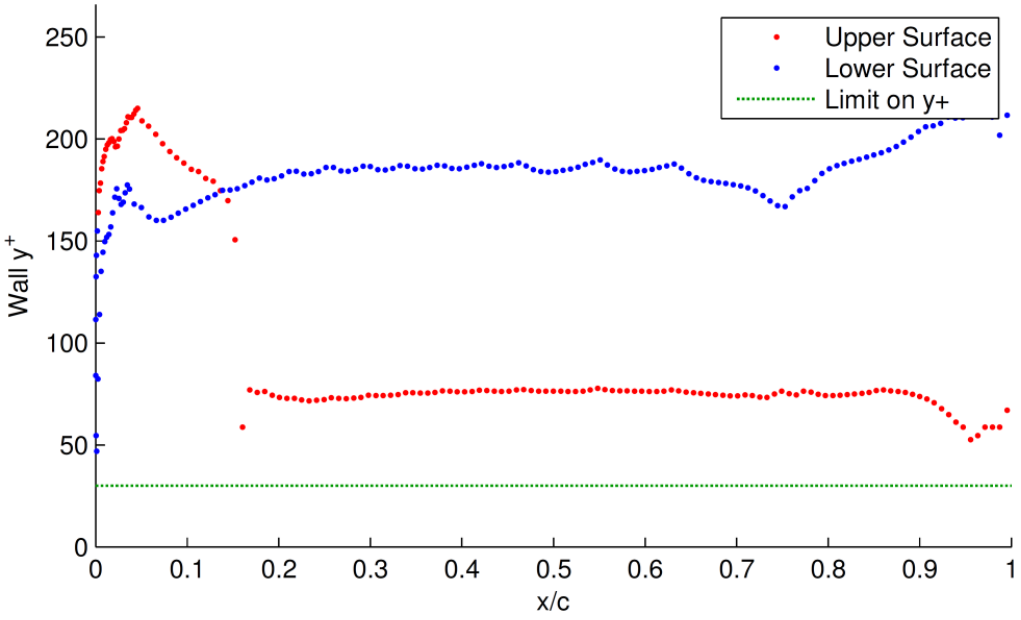


Figure D.44: Wall $y+$ for $\alpha = 9°$, $M = 0.8$

Appendices E PABLO .m Files

This Appendix contains the PABLO m-files that were modified for use in the optimisation procedure. Refer to §6.1 to §6.7 for explanations.

E.1 pablo.m

```
function [c_l c_h_a c_h_t] = pablo(airfoil,x_a,y_a,x_t,y_t,...
                                   alpha,delta_a,delta_t)

panels = 280;   % number of vortex element panels used for computation

% create profile with desired panels and control surface deflections
z = create_profile(panels/2,airfoil,delta_a,delta_t,x_a,y_a,x_t,y_t);

% calculate pressure distribution
[ue, Cp] = vortex(z,alpha);

% coeffificient of lift
c_l = lift(Cp,z,alpha)

% hinge moment coefficient
c_h_a = moment(Cp,z,x_a,y_a);
c_h_t = moment(Cp,z,x_t,y_t);
```

E.2 create_profile.m

```
function z = create_profile(nbpo2,airfoil,delta_a,delta_t,x_a,y_a,x_t,y_t)

% Read in profile data from selected file
filename = ([airfoil '.data']);
fid = fopen(filename, 'rt');
fgetl(fid); % discard first line
i = 0;
while 1
    i = i+1;
    characters = fgetl(fid);
    if ~ischar(characters), break, end
    data(i,:) = str2num(characters);
end
fclose(fid);
xdat = data(:,1);
ydat = data(:,2);

% Find LE and split data into upper and lower profiles
[h i] = min(xdat);
```

```
20:     xdatup = xdat(i:-1:1);
21:     ydatup = ydat(i:-1:1);
22:     xdatlo = xdat(i+1:end);
23:     ydatlo = ydat(i+1:end);
24:
25:     % create new x vector with higer distribution over flap hinges
26:     x_up = distribute_x(delta_a, delta_t, x_a, x_t, nbpo2);
27:     x_lo = x_up;
28:
29:     % Parameterise the existing arc length and spline profile to get yup & ylo
30:     sup = [0,cumsum(sqrt(diff(xdatup,1,1)'.^2+diff(ydatup,1,1)'.^2))];
31:     slo = [0,cumsum(sqrt(diff(xdatlo,1,1)'.^2+diff(ydatlo,1,1)'.^2))];
32:
33:     % Upper surface
34:     Vxc = sqrt(x_up);
35:     sVxc = interp1(sqrt(xdatup),sup,Vxc,'spline');
36:     y_up = spline(sup,ydatup,sVxc);
37:     y_up(1) = 0; % this should already be 0, but it goes wonky sometimes...
38:
39:     % Lower surface
40:     Vxc = sqrt(x_lo);
41:     sVxc = interp1(sqrt(xdatlo),slo,Vxc,'spline');
42:     y_lo = spline(slo,ydatlo,sVxc);
43:
44:     % close TE gap
45:     upper_error = y_up(end);
46:     for i = 1:size(y_up)
47:       y_up(i) = y_up(i) - upper_error*x_up(i);
48:     end
49:     lower_error = y_lo(end);
50:     for i = 1:size(y_lo)
51:       y_lo(i) = y_lo(i) - lower_error*x_lo(i);
52:     end
53:
54:     % adjust profile for control surface deflections
55:     if delta_a ~= 0 && delta_t ~= 0 % most likely scenario
56:         % rotate tab profile first
57:         [x_up y_up x_lo y_lo] = ...
58:                         rotate_hinge(delta_t,x_t,y_t,x_up,x_lo,y_up,y_lo);
59:         % rotate aileron profile
60:         [x_up y_up x_lo y_lo] = ...
61:                         rotate_hinge(delta_a,x_a,y_a,x_up,x_lo,y_up,y_lo);
62:     elseif delta_a ~= 0 % delta_t will be zero, rotate aileron only
63:         [x_up y_up x_lo y_lo] = ...
64:                         rotate_hinge(delta_a,x_a,y_a,x_up,x_lo,y_up,y_lo);
65:     elseif delta_t ~= 0 % delta_a will be zero, rotate tab only
66:         [x_up y_up x_lo y_lo] = ...
67:                         rotate_hinge(delta_t,x_t,y_t,x_up,x_lo,y_up,y_lo);
68:     else % delta_a = 0 & delta_t = 0
69:         % no change
70:     end
71:
72:     % Ensure chord length is unity:
73:     chord = sqrt((x_up(end)-x_up(1))^2+(y_up(end)-y_up(1))^2);
74:     % adjust x so that c = 1
75:     xdatup = x_up/chord;
76:     xdatlo = x_lo/chord;
77:     % adjust y similarly to prevent distortion
78:     ydatup = y_up/chord;
79:     ydatlo = y_lo/chord;
80:
81:     %% Refresh Profile
82:     % refresh node redistribution
83:     x_up = distribute_x(delta_a, delta_t, x_a, x_t, nbpo2);
84:     x_lo = x_up;
85:
```

```
86:     % Parameterise the existing arc length and spline profile to get yup & ylo
87:     sup = [0,cumsum(sqrt(diff(xdatup,1,1)'.^2+diff(ydatup,1,1)'.^2))];
88:     slo = [0,cumsum(sqrt(diff(xdatlo,1,1)'.^2+diff(ydatlo,1,1)'.^2))];
89:
90:     % Upper surface
91:     Vxc = sqrt(x_up);
92:     sVxc = interp1(sqrt(xdatup),sup,Vxc,'spline');
93:     y_up = spline(sup,ydatup,sVxc);
94:     y_up(1) = 0;
95:
96:     % Lower surface
97:     Vxc = sqrt(x_lo);
98:     sVxc = interp1(sqrt(xdatlo),slo,Vxc,'spline');
99:     y_lo = spline(slo,ydatlo,sVxc);
100:
101:    % arrange data for output
102:    z = [x_up(end:-1:1) y_up(end:-1:1); x_lo(2:end) y_lo(2:end)];
```

E.3 distribute_x.m

```
1:    function x = distribute_x(delta_a, delta_t, x_a, x_t, nbpo2)
2:
3:    % x distribution
4:     if delta_a == 0 && delta_t == 0 % no deflections %
5:        beta = linspace(0,pi,nbpo2);
6:        x = 0.5*(1-cos(beta))';
7:    elseif delta_a ~= 0 && delta_t == 0 % aileron deflection only
8:    flap_node_weight = 1.5; % proportion of extra nodes after hinge
9:        tail_nodes = (1-x_a)*flap_node_weight;
10:       main_nodes = 1-tail_nodes;
11:       beta_1 = linspace(0,pi,round(nbpo2*main_nodes));
12:       beta_2 = linspace(0,pi,round(nbpo2*tail_nodes));
13:       x_1 = 0.5*x_a*(1-cos(beta_1))';
14:       x_2 = 0.5*(1-x_a)*(1-cos(beta_2))';
15:       x = [x_1; x_1(end)+x_2(2:end)];
16:   elseif delta_a == 0 && delta_t ~=0 % tab deflection only
17:       flap_node_weight = 3; % proportion of extra nodes after hinge
18:       tail_nodes = (1-x_t)*flap_node_weight;
19:       main_nodes = 1-tail_nodes;
20:      beta_1 = linspace(0,pi,nbpo2*main_nodes);
21:       beta_2 = linspace(0,pi,nbpo2*tail_nodes);
22:       x_1 = 0.5*x_t*(1-cos(beta_1))';
23:       x_2 = 0.5*(1-x_t)*(1-cos(beta_2))';
24:       x = [x_1; x_1(end)+x_2(2:end)];
25:   else % aileron and tab deflections
26:       aileron_node_weight = 1.5;
27:       tab_node_weight = 3;
28:       tab_nodes = (1-x_t)*tab_node_weight;
29:       aileron_nodes = (x_t-x_a)*aileron_node_weight;
30:       main_nodes = 1 - tab_nodes - aileron_nodes;
31:       beta_1 = linspace(0,pi,nbpo2*main_nodes);
32:       beta_2 = linspace(0,pi,nbpo2*aileron_nodes);
33:       beta_3 = linspace(0,pi,nbpo2*tab_nodes);
34:       x_1 = 0.5*x_a*(1-cos(beta_1))';
35:       x_2 = 0.5*(x_t-x_a)*(1-cos(beta_2))';
36:       x_3 = 0.5*(1-x_t)*(1-cos(beta_3))';
37:       x = [x_1; x_1(end)+x_2(2:end); x_1(end)+x_2(end)+x_3(2:end)];
38:    end
```

E.4 rotate_hinge.m

```
% This  function simply rotates the nodes aft of the hinge in question by
% the desired deflection angle. The point where the panels intersection is
% added to the existing nodes.
% The tab must be rotated before the aileron to ensure the correct geometry
% results.

% Built from scratch by Chris Day, June 2009

function [xu yu xl yl] = rotate_hinge(delta,x_h,y_h,xu,xl,yu,yl)

% convert flap deflection to radians
delta = delta*pi/180;

% find index of nodes on either side of hinge
index_i = find(xu < x_h, 1, 'last' );
index_j = index_i+1;
index_k = find(xl < x_h, 1, 'last' );
index_l = index_j+1;
% interpolate to find where profiles intersect vertical line at x = x_h
y1 = (x_h-xu(index_i))*(yu(index_j)-yu(index_i))/...
    (xu(index_j)-xu(index_i)) + yu(index_i);
y2 = (x_h-xl(index_k))*(yl(index_l)-yl(index_k))/...
    (xl(index_l)-xl(index_k)) + yl(index_k);
% calculate desired absolute y-position of hinge point
y_h = y1 + y_h*(y2 - y1);

% move hinge to origin (and create separate upper and lower x vectors)
xu_centred = xu - x_h;
yu_centred = yu - y_h;
xl_centred = xl - x_h;
yl_centred = yl - y_h;
% rotation about origin (clockwise positive)
% [x'; y'] = [x; y][cos(deltat) sin(deltat); -sin(deltat) cos(deltat)];
xu_centred = xu_centred*cos(delta) + yu_centred*sin(delta);
yu_centred = -xu_centred*sin(delta) + yu_centred*cos(delta);
xl_centred = xl_centred*cos(delta) + yl_centred*sin(delta);
yl_centred = -xl_centred*sin(delta) + yl_centred*cos(delta);
% move profile back from origin (and restore previous nomenclature)
xu_flapped = xu_centred + x_h;
yu_flapped = yu_centred + y_h;
xl_flapped = xl_centred + x_h;
yl_flapped = yl_centred + y_h;

% join profiles 1 - upper profile:
% find point of intersection
[xu_join,yu_join] = intersections(xu,yu,xu_flapped,yu_flapped);
% split each profile and join, adding point of intersection
xu_lead = xu(xu<xu_join);
yu_lead = yu(xu<xu_join);
xu_trail = xu_flapped(xu_flapped>xu_join);
yu_trail = yu_flapped(xu_flapped>xu_join);
xu = [xu_lead; xu_join; xu_trail];
yu = [yu_lead; yu_join; yu_trail];

% join profiles 2 - lower profile:
% find point of intersection
[xl_join,yl_join] = intersections(xl,yl,xl_flapped,yl_flapped);
% split each profile and join, adding point of intersection
xl_lead = xl(xl<xl_join);
yl_lead = yl(xl<xl_join);
xl_trail = xl_flapped(xl_flapped>xl_join);
```

```
62:     yl_trail = yl_flapped(xl_flapped>xl_join);
63:     xl = [xl_lead; xl_join; xl_trail];
64:     yl = [yl_lead; yl_join; yl_trail];
65:
```

E.5 vortex.m

```
1:     function cp = vortex(za,alpha)
2:
3:     % This file was slightly edited to return only the pressure distributions,
4:     % as opposed to a slew of other unnecessary characteristics. The required
5:     % charactristics are now computed from the pressure distribution in the
6:     % parent function. The original function header appears below.
7:
8:     % This program finds and plots the pressure distribution on an airfoil by
9:     % representing the surface as a finite number of linear strength vortex
10:    % panels. (Neumann Boundary condition V.n = 0)
11:    % Airfoils are taken from the Naca 4 digits library.
12:
13:    % Input data
14:    % za is an array containing the airfoil panels coordinates
15:    % alpha is the angle of attack expressed in degrees
16:
17:    nbp = max(size(za))-1; % number of panels
18:    chord = 1;
19:    Vzero = 1;
20:    alphar = pi.*alpha./180;
21:
22:    % Turn z into an array of complex number
23:    z = za(:,1)+i*za(:,2);
24:
25:    % Change z to clockwise
26:    z = z(nbp+1:-1:1);
27:
28:    % Collocation points
29:    m =(z(1:nbp)+z(2:nbp+1))/2;
30:
31:    % Panel angle
32:    th = imag(log(z(2:nbp+1)-z(1:nbp)));
33:
34:    % Free stream normal velocity component
35:    RHSi(1:nbp,1) = cos(alphar)*sin(th(1:nbp))-sin(alphar)*cos(th(1:nbp));
36:
37:    % Influence matrix
38:    % convert collocation pt to panel coordinate system
39:    xzt = m*ones(1,nbp)-ones(nbp,1)*z(1:nbp).';
40:    xt = real(xzt);
41:    zt = imag(xzt);
42:
43:    xz2t = diff(z);
44:    x2t = real(xz2t);
45:    z2t = imag(xz2t);
46:
47:    cth = ones(nbp,1)*cos(th).';
48:    sth = ones(nbp,1)*sin(th).';
49:
50:    X = xt.*cth+zt.*sth;
51:    Z = -xt.*sth+zt.*cth;
52:    X2 = x2t.*cos(th)+z2t.*sin(th);
53:
54:    % compute r1,r2 and th2-th1
```

```
55:     mii = m*ones(1,nbp);
56:     zjj = ones(nbp,1)*z(1:nbp).';
57:     zjjp1 = ones(nbp,1)*z(2:nbp+1).';
58:
59:     r1 = abs(zjj-mii);
60:     r2 = abs(zjjp1-mii);
61:
62:     angle = imag(log((zjjp1-mii)./(zjj-mii))) ;
63:     angle = mod(angle-pi,2*pi)-pi;
64:     tmp = X2(:);
65:     X2mat = ones(nbp,1)*tmp';
66:
67:     th2mth1 = angle./(2*pi*X2mat);
68:     RR = log(r2./r1)./(2*pi*X2mat);
69:     u2l = (Z.*RR+X.*th2mth1);
70:     u1l = -(u2l-X2mat.*th2mth1);
71:     cnst = 1/(2*pi);
72:     TMP = 1/(2*pi)-Z.*th2mth1;
73:     w1l = -TMP+(X2mat-X).*RR;
74:     w2l = TMP+X.*RR;
75:
76:     tmp = diag(u1l) + 0.5*( diag(X)-X2  )./X2;
77:     u1l = u1l - diag( tmp );
78:     tmp = diag(u2l) - 0.5*diag(X)./X2;
79:     u2l = u2l - diag(tmp);
80:     tmp = diag(w1l);
81:     w1l = w1l - diag(tmp + 1/(2*pi));
82:     tmp = diag(w2l);
83:     w2l = w2l - diag(tmp - 1/(2*pi));
84:
85:     % Velocity in global coordinate system
86:     ca = ones(nbp,1)*cos(-th)';
87:     sa = ones(nbp,1)*sin(-th)';
88:
89:     u1 = u1l.*ca+w1l.*sa;
90:     u2 = u2l.*ca+w2l.*sa;
91:     w1 = -u1l.*sa+w1l.*ca;
92:     w2 = -u2l.*sa+w2l.*ca;
93:
94:     % Influence matrix coefficient
95:     CA = cos(th)*ones(1,nbp);
96:     SA = sin(th)*ones(1,nbp);
97:     Aij = zeros(nbp,nbp+1);
98:     Bij = Aij;
99:     Aij(:,1:nbp) = -u1.*SA + w1.*CA;
100:    Bij(:,1:nbp) =  u1.*CA + w1.*SA;
101:    Aij(:,2:nbp+1) = Aij(:,2:nbp+1) - u2.*SA + w2.*CA;
102:    Bij(:,2:nbp+1) = Bij(:,2:nbp+1) + u2.*CA + w2.*SA;
103:
104:    % Add a wake panel with a constant-strength vortex
105:    % Infinite wake point
106:    d1n = dist(za,nbp,nbp+1);
107:    d12 = dist(za,2,1);
108:    xP = (za(nbp,1)-za(nbp+1,1))./d1n + (za(2,1)-za(1,1))./d12;
109:    yP = (za(nbp,2)-za(nbp+1,2))./d1n + (za(2,2)-za(1,2))./d12;
110:    dPo = sqrt(xP.^2+yP.^2);
111:    zi1 = ((za(1,1)+za(nbp+1,1))/2-chord.*2000.*xP./dPo);
112:    zi2 = ((za(1,2)+za(nbp+1,2))/2-chord.*2000.*yP./dPo);
113:
114:    zi = zi1 + i*zi2;
115:    zte = z(1);
116:
117:    d1 = zte*ones(nbp,1) - m;
118:    d2 = zi*ones(nbp,1) - m;
119:
120:    angle = imag(log(d1./d2)) ;
```

```
121:     angle = mod(angle-pi,2*pi)-pi;
122:
123:     r1or2 = abs(d1)./abs(d2);
124:
125:     u = 1/(2*pi)*angle;
126:     w = -1/(2*pi)*log(r1or2);
127:
128:     % transfer to global coordinate system
129:     ca = cos(-th);
130:     sa = sin(-th);
131:
132:     ug = u.*ca+w.*sa;
133:     wg = -u.*sa+w.*ca;
134:
135:     % find the tangential component :
136:     CA = cos(th);
137:     SA = sin(th);
138:     Aw = -ug.*SA + wg.*CA;
139:     Aij = [Aij Aw];
140:
141:     % Kutta condition
142:     Aij(nbp+1,1)     = 1;
143:     Aij(nbp+2,nbp+1) = 1;
144:
145:     RHSi(nbp+1,1) = 0;
146:     RHSi(nbp+2,1) = 0;
147:
148:     % Solve
149:     gamma = Aij\RHSi;
150:     gamma = gamma(1:nbp+1,1);
151:
152:     % Compute velocity
153:     vel =  Bij*gamma;
154:     velocity = vel + cos(alphar)*cos(th)+sin(alphar)*sin(th);
155:
156:     % turn velocity to anti-clockwise
157:     velocity = -velocity(nbp:-1:1);
158:
159:     % compute cp at collocation points
160:     cp = 1-velocity.^2/Vzero.^2;
```

E.6 lift.m

```
1:     function c_l = lift(Cp,z,alpha)
2:
3:     alpha = alpha*pi/180;
4:
5:     % panel angle of inclination to x axis
6:     theta = atan2(z(1:end-1,2)-z(2:end,2),z(1:end-1,1)-z(2:end,1));
7:
8:     % panel length
9:     ds = sqrt((z(1:end-1,2)-z(2:end,2)).^2+(z(1:end-1,1)-z(2:end,1)).^2);
10:
11:    % the lift coefficient
12:    c_l = -trapz(Cp.*ds.*cos(theta-alpha));
```

E.7 moment.m

```
% This function calculates the moment of the pressure distribution
% only, about any point in space.

% It requires the pressure distribution and profile's nodal
% coordinates, as well as the position of the hinge axis.

% Created from scratch by Chris Day. Last modified 27 August 2009.

function c_m = moment(Cp,z,x_h,y_h)

% panel angle of inclination to x axis
theta = atan2(z(1:end-1,2)-z(2:end,2),z(1:end-1,1)-z(2:end,1));

% panel length
ds = sqrt((z(1:end-1,2)-z(2:end,2)).^2+(z(1:end-1,1)-z(2:end,1)).^2);

% panel midpoint
x_mid = z(1:end-1,1)+(z(2:end,1)-z(1:end-1,1))/2;
y_mid = z(1:end-1,2)+(z(2:end,2)-z(1:end-1,2))/2;

% remove pressure values to the left of the hinge point
Cp(x_mid<x_h) = 0;

% calculate absolute y location of hinge
x = z(:,1); y = z(:,2);                    % split into x and y
halfway = round(length(x)/2);
xu = x(1:halfway); xl = x(halfway:end); % split roughly into up and low
yu = y(1:halfway); yl = y(halfway:end);
disp(['x_h = ' num2str(x_h)])
index_i = find(xu > x_h, 1, 'last' );   % find nodes of panel above hinge
index_j = index_i+1;
index_k = find(xl < x_h, 1, 'last' );   % find nodes of panel below hinge
index_l = index_k+1;                    % interpolation...
y1 = (x_h-xu(index_i))*(yu(index_j)-yu(index_i))/...
    (xu(index_j)-xu(index_i)) + yu(index_i);
y2 = (x_h-xl(index_k))*(yl(index_l)-yl(index_k))/...
    (xl(index_l)-xl(index_k)) + yl(index_k);
y_h = y1 + y_h*(y2 - y1);               % absolute y location of hinge

% aaaaand... the moment we've all been waiting for!
c_m = trapz(Cp.*ds.*((x_mid-x_h).*cos(theta) + (y_mid-y_h).*sin(theta)));
```

Appendices F Matlab Optimisation Code

This appendix contains the code used to generate test arrays and responses for analysis in Design Expert. The code in §F.1 was used for generation of data for the full-factorial analysis, while the code in §F.2 to §F.6 was used to generate output for the fractional-factorial analysis. Explanations appear before presentation of the code in each section, summarising the logic where necessary or relevant.

F.1 `generate_output_full.m`

In this script an array is created having rows with every combination of factors (23–38). The test matrix is set up as a cell array (43) to enable insertion of both numeric and character data, as Design Expert uses the actual factor values. These are recorded in a loop (45–53) referring to the factor levels in the array to find the actual values, and which also generates each response by calling PABLO for the given factor levels for each run. Finally, the results are exported to Excel (56–58) to facilitate insertion into Design Expert.


```
% This script creates a full-factorial test array and inserts factors in
% the order as listed in lines 15 to 20. The array is sent to PABLO one
% configuration at a time, yielding responses for C_L, C_Ha and C_Ht, which
% are recorded. The test array and responses are ouput to Excel for easy
% insertion into Design Expert via Copy & Paste.

% Written by Chris Day, 14 September 2010

%% Clean the slate
clc
clear all
close all

%% Define control factors and their levels
alpha = [-4 0 4];                       % factor 1
delta_a = [-4 0 4];                     % factor 2
delta_t = [-4 0 4];                     % factor 3
x_a = [0.74 0.76 0.78];                 % factor 4
x_t = [0.9324 0.9424 0.9516];           % factor 5
airfoil = {'00012' '23012' '23018'};    % factor 6
num_factors = 6;

%% Set up full-factorial array for all 6 factors
OA = zeros(3^num_factors,num_factors); i = 0;
for A =1:length(alpha)
  for B =1:length(delta_a)
    for C =1:length(delta_t)
      for D =1:length(x_a)
```

```
29:             for E =1:length(x_t)
30:               for F =1:length(airfoil)
31:                 i = i+1;
32:                 OA(i,:) = [A B C D E F];
33:               end
34:             end
35:           end
36:         end
37:       end
38:     end
39:     [n,f] = size(OA);
40:
41:     %% Generate responses and test matrix
42:     cl = NaN(length(OA),1); cha = cl; cht = cl;
43:     test_matrix{n,f} = [];
44:     for i = 1:n
45:       [cl(i) cha(i) cht(i)] = pablo(airfoil{OA(i,6)},...
46:                   x_a(OA(i,4)),0.5x_t(OA(i,5)),0.5,,...
47:                   alpha(OA(i,1)), delta_a(OA(i,2)),delta_t(OA(i,3)));
48:       test_matrix(i,:) = {...
49:                   alpha(OA(i,1)) delta_a(OA(i,2)) delta_t(OA(i,3)) ...
50:                   x_a(OA(i,4)) x_t(OA(i,5))  airfoil{OA(i,6)}};
51:     end
52:
53:     %% Export for Design Expert 7 insertion
54:     xlswrite('full_test_matrix.xlsx', test_matrix, 'DX7 Array', 'A2')
55:     xlswrite('full_test_matrix.xlsx', [cl cha cht], 'DX7 Array', 'G2')
```

F.2 generate_output_L81.m

This script controls the whole process of factor column assignment for the L_{81} array. After the factors are defined (11–17), the required array is read in (20, see §F.6). The factors and interactions of interest are then listed (23–25). The function `assign_factors` is then called (26, see §F.3), which returns the columns of the array to which each factor must be assigned. The responses are then recorded by sending each configuration to PABLO, and a test matrix is generated with actual factor values (40–53). These are then exported to Excel to facilitate insertion into Design Expert (55–58).


```
1:    % This script loads an appropriate orthogonal array, assigns factors
2:    % to the correct columns according to the interactions under examination,
3:    % runs Pablo to yield responses for each treatment combination, and outputs
4:    % the test matrix for insertion into Design Expert.
5:
6:    % Written by Chris Day, 19 October 2010
7:
8:    clc
9:    clear all
10:
11:   %% Factors and their levels
12:   alpha = [-4 0 4];                     % A
13:   delta_a = [-4 0 4];                   % B
14:   delta_t = [-4 0 4];                   % C
15:   x_a = [0.74 0.76 0.78];               % D
16:   x_t = [0.9324 0.9424 0.9516];         % E
17:   airfoil = {'00012' '23012' '23018'};  % F
18:
19:   %% Load orthogonal array, generate interaction table and get column assignments
```

```
20:     L81
21:
22:     % factors and interactions of interest
23:     factors = 'ABCDEF';
24:     interactions = ['AB' 'AC' 'AD' 'AE' 'AF' 'BC' 'BD' 'BE' 'BF' 'CD' 'CE' 'CF'...
25:                     'DE' 'DF' 'EF'];
26:     location = assign_factors(OA,factors,interactions);
27:
28:     %% Column assignments for each factor
29:     A = location(1);
30:     B = location(2);
31:     C = location(3);
32:     D = location(4);
33:     E = location(5);
34:     F = location(6);
35:
36:     %% Test matrix and response for Design Expert analysis
37:     n = length(OA);
38:     f = length(factors);
39:
40:     %% Generate responses from test matrix
41:     cl = NaN([length(OA),1]); cha = cl; cht = cl;
42:     test_matrix{n,f} = [];
43:     h = waitbar(0,'Calculating response...');
44:     for i = 1:n
45:      waitbar(i/n)
46:         [cl(i) cha(i) cht(i)] = pablo(...
47:                     airfoil{OA(i,F)},x_a(OA(i,D)),0.5,x_t(OA(i,E)),0.5, ...
48:                     alpha(OA(i,A)), delta_a(OA(i,B)),delta_t(OA(i,C)));
49:      test_matrix(i,:) = {...
50:                     alpha(OA(i,A)) delta_a(OA(i,B)) delta_t(OA(i,C)) ...
51:         x_a(OA(i,D)) x_t(OA(i,E))  airfoil{OA(i,F)}};
52:     end
53:     close(h);
54:
55:     %% Export for Design Expert 7 insertion
56:     xlswrite('L81_test_matrix_and_response.xlsx', test_matrix, 'DX7 Array', 'A2')
57:     xlswrite('L81_test_matrix_and_response.xlsx', [cl cha cht], 'DX7 Array', 'G2')
58:     disp('Results successfully output to Excel.')
```

F.3 assign_factors.m

The function `assign_factors` assigns factors to columns of the specified array. To set up the process an interaction table is generated from the orthogonal array itself (13–40) (see §F.4). Factors are then randomly assigned to columns. The columns of each interacting pair are then recorded (21–38), and a list of confounded columns is created (39–40) using the function `check_confounding` (see §F.5). The process of randomly assigning factors to columns continues until a solution is found.

Although this function is computationally inefficient, a solution is generally found within a second.

```
1:  % This script randomly assigns factors to columns of an L81 fractional-
2:  % factorial test array and checks for confounding according to the
3:  % interactions listed in "interaction_string". When both factors and
4:  % interactions are found to be free of confounding the solution is output
5:  % to the command prompt.
6:
7:  % Written by Chris Day, 15 July 2011
8:
9:  function location = assign_factors(OA,factors,interaction_string)
10:
11: % get interaction table
12: IT = interaction_table(OA);
13:
14: %  assign factors to columns
15: solution = false;
16: while solution == false
17:     % create random order of i = 1:40
18:     i = randperm(40);
19:     % place factors in first columns
20:     location = i(1:6);
21:     % calculate columns of interactions to be studied
22:     interaction = NaN(1,length(interaction_string));
23:     for i = 1:length(interaction_string)
24:         switch interaction_string(i)
25:             case 'A'
26:                 interaction(i) = location(1);
27:             case 'B'
28:                 interaction(i) = location(2);
29:             case 'C'
30:                 interaction(i) = location(3);
31:             case 'D'
32:                 interaction(i) = location(4);
33:             case 'E'
34:                 interaction(i) = location(5);
35:             case 'F'
36:                 interaction(i) = location(6);
37:         end
38:     end
39:     % check confounded columns
40:     confounded = check_confounding(interaction,IT);
41:
42:     % check if facors or interactions are confounded
43:     if ~max(ismember(location,confounded))
44:         solution = true;
45:     end
46: end
```

F.4 interaction_table.m

This function was written more as an exercise than because it was absolutely necessary. Published interaction tables exist, and this algorithm is restricted to arrays having only 3-level factors. As such, it will only work with the L_9, L_27 and L_{81} orthogonal arrays having entries designated by -1, 0, and 1. The interaction table of the L_{81} array is 2 pages of 40 rows by 40 columns, so the main initial purpose of this algorithm was to avoid transcription errors and the tedium of copying the table, although it required a working knowledge of why confounding occurs.

An interaction table for 3-level factors consists of two 2D arrays. Each interaction will have two confounded columns, and both arrays must be consulted to find the corresponding entries.

The process of building the table starts with the property of linear dependence of the columns. In the first of a four-stage process, all linearly dependent columns are found and recorded. This is done by checking each unique combination of 3 columns (16–19), that for columns designated a, b, and c, $c \neq a+b$ (21). If $c = a+b$ then column c will be confounded with an interaction of factors in column a and b.

The interaction table must be filled in two passes. The results from above are used to fill the table (34–49). However, even though all the information has been inputted to the table, empty cells will exist as the interaction table itself contains duplicates. A second pass is therefore conducted (51-70) where if any cells that did not receive entries are found (58–60) and corrected (61–66).

The last stage involves simply placing the entries, which up to now have existed as two single working arrays, into a single 3D (2-page) array (72–81).


```
 1:  % This function uses an orthogonal array having only 3-level factors and
 2:  % generates an interaction table as a 2-page (3D) array.
 3:
 4:  % Written by Chris Day, 19 October 2010
 5:
 6:  function IT = interaction_table(OA)
 7:
 8:  % convert array to working format (elements must be 0, 1 & 2)
 9:  OA = OA + (2 - max(max(OA)));
10:
11:  % 1: find linearly dependant columns
12:  [~, n] = size(OA);
13:  % create container for linear dependent columns found
14:  LD{n,n} = [];
15:  % find linearly dependant columns
16:  for a = 1:n
17:      for b = 1:n
18:          for c = 1:n
19:              if a == b || a == c || b == c
20:                  % skip / do nothing - interaction not defined/possible
21:              elseif nnz(mod(OA(:,a)+OA(:,b),3)-OA(:,c)) == 0
22:                  % nnz (above) = calculates number of non zero elements.
23:                  % if nnz(x) = 0, all elements of x are zero, therefore
```

```
                    % if x = a + b - c, a + b = c.
                    % record linear dependence
                    LD{a,b} = [LD{a,b} c];
                    LD{a,c} = [LD{a,c} b];
                    LD{b,c} = [LD{b,c} a];
                end
            end
        end
    end

    % 2: Pass 1 - fill interaction table from linearly dependant columns
    IT1 = zeros(n,n); IT2 = zeros(n,n);
    for i = 1:n
        for j = 1:n
            if i <= j
                % extract two confounded columns numbers
                confounded = unique(LD{i,j});
                if numel(confounded) > 0
                    IT1(i,j) = confounded(1);
                end
                if numel(confounded) > 1
                    IT2(i,j) = confounded(2);
                end
            end
        end
    end

    % 3: Pass 2 - complete interaction table from the interaction table itself
    IT_temp{n,n} = [];
    for i = 1:n
        for j = 1:n
            if i < j
                k = IT1(i,j);
                l = IT2(i,j);
                confounded = sort([i j k l]);
                % if i|j|k|l = 0 interaction was not picked up properly
                if confounded(1) ~= 0
                    IT_temp{i,j} = [k l];
                    IT_temp{i,k} = [j l];
                    IT_temp{i,l} = [j k];
                    IT_temp{j,k} = [i l];
                    IT_temp{j,l} = [i k];
                    IT_temp{k,l} = [i j];
                end
            end
        end
    end

    % 4: create two-page interaction tables
    IT = zeros(n,n,2);
    for i = 1:n
        for j = 1:n
            if i < j
                IT(i,j,1) = IT_temp{i,j}(1);
                IT(i,j,2) = IT_temp{i,j}(2);
            end
        end
    end
```

F.5 check_confounding.m

This function refers to the interaction table generated in §F.4, and checks the confounded columns of two factors that may be involved in an interaction. The inputted string of interactions are read in pair by pair(4–7), with each pair in numerical order (8–13). The returned variable is filled with columns which are to be avoided, as read from the interaction table (14–15).

```
1:  function confounded = check_confounding(interaction,IT)
2:
3:  confounded = NaN(1,length(interaction));
4:  % calculate two confounded columns for each interaction
5:  for i = 1:length(interaction)/2
6:     column_1 = interaction(2*i-1);
7:     column_2 = interaction(2*i);
8:     % ensure column_1 < column_2
9:     if column_1 > column_2
10:       temp = column_1;
11:       column_1 = column_2;
12:       column_2 = temp;
13:    end
14:    confounded(2*i-1) = IT(column_1,column_2,1);
15:    confounded(2*i) = IT(column_1,column_2,2);
16: end
```

F.6 L81.m

This file has been included for completeness, and simply contains the $L_{81}\,(3^{40})$ orthogonal array.

```
% L_81 (3^40) orthogonal array
OA = [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
      1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
      1 1 1 1 1 1 1 1 1 1 1 1 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
      1 1 1 1 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3
      1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 1 1 1 1 1 1 1 1
      1 1 1 1 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2
      1 1 1 1 3 3 3 3 3 3 3 3 3 1 1 1 1 1 1 1 1 1 3 3 3 3 3 3 3 3 3 3 2 2 2 2 2 2 2 2
      1 1 1 1 3 3 3 3 3 3 3 3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 3 3 3 3 3 3 3 3
      1 1 1 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 1 1 1 1 1 1 1
      1 2 2 2 1 1 1 2 2 2 3 3 3 1 1 1 2 2 2 3 3 3 1 1 1 2 2 2 3 3 3 1 1 1 2 2 2 3 3 3
      1 2 2 2 1 1 1 2 2 2 3 3 3 2 2 2 3 3 3 1 1 1 2 2 2 3 3 3 1 1 1 2 2 2 3 3 3 1 1 1
      1 2 2 2 1 1 1 2 2 2 3 3 3 3 3 3 1 1 1 2 2 2 3 3 3 1 1 1 2 2 2 3 3 3 1 1 1 2 2 2
      1 2 2 2 2 2 2 3 3 3 1 1 1 1 1 1 2 2 2 3 3 3 2 2 2 3 3 3 1 1 1 2 2 2 3 3 3 1 1 1
      1 2 2 2 2 2 2 3 3 3 1 1 1 2 2 2 3 3 3 1 1 1 3 3 3 1 1 1 2 2 2 1 1 1 2 2 2 3 3 3
      1 2 2 2 2 2 2 3 3 3 1 1 1 3 3 3 1 1 1 2 2 2 1 1 1 2 2 2 3 3 3 2 2 2 3 3 3 1 1 1
      1 2 2 2 3 3 3 1 1 1 2 2 2 1 1 1 2 2 2 3 3 3 3 3 3 1 1 1 2 2 2 2 2 2 3 3 3 1 1 1
      1 2 2 2 3 3 3 1 1 1 2 2 2 2 2 2 3 3 3 1 1 1 1 1 1 2 2 2 3 3 3 3 3 3 1 1 1 2 2 2
      1 2 2 2 3 3 3 1 1 1 2 2 2 3 3 3 1 1 1 2 2 2 2 2 2 3 3 3 1 1 1 1 1 1 2 2 2 3 3 3
      1 3 3 3 1 1 1 3 3 3 2 2 2 1 1 1 3 3 3 2 2 2 1 1 1 3 3 3 2 2 2 1 1 1 3 3 3 2 2 2
      1 3 3 3 1 1 1 3 3 3 2 2 2 2 2 2 1 1 1 3 3 3 2 2 2 1 1 1 3 3 3 2 2 2 1 1 1 3 3 3
      1 3 3 3 1 1 1 3 3 3 2 2 2 3 3 3 2 2 2 1 1 1 3 3 3 2 2 2 1 1 1 3 3 3 2 2 2 1 1 1
      1 3 3 3 2 2 2 1 1 1 3 3 3 1 1 1 3 3 3 2 2 2 2 2 2 1 1 1 3 3 3 3 3 3 2 2 2 1 1 1
      1 3 3 3 2 2 2 1 1 1 3 3 3 2 2 2 1 1 1 3 3 3 3 3 3 2 2 2 1 1 1 1 1 1 3 3 3 2 2 2
      1 3 3 3 2 2 2 1 1 1 3 3 3 3 3 3 2 2 2 1 1 1 1 1 1 3 3 3 2 2 2 2 2 2 1 1 1 3 3 3
      1 3 3 3 3 3 3 2 2 2 1 1 1 1 1 1 3 3 3 2 2 2 3 3 3 2 2 2 1 1 1 2 2 2 1 1 1 3 3 3
      1 3 3 3 3 3 3 2 2 2 1 1 1 2 2 2 1 1 1 3 3 3 1 1 1 3 3 3 2 2 2 3 3 3 2 2 2 1 1 1
      1 3 3 3 3 3 3 2 2 2 1 1 1 3 3 3 2 2 2 1 1 1 2 2 2 1 1 1 3 3 3 1 1 1 3 3 3 2 2 2
      2 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3
      2 1 2 3 1 2 3 1 2 3 1 2 3 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1
```

```
31:     2 1 2 3 1 2 3 1 2 3 1 2 3 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2
32:     2 1 2 3 2 3 1 2 3 1 2 3 1 1 2 3 1 2 3 1 2 3 2 3 1 2 3 1 2 3 1 3 1 2 3 1 2 3 1 2
33:     2 1 2 3 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 3 1 2 3 1 2 3 1 2 1 2 3 1 2 3 1 2 3
34:     2 1 2 3 2 3 1 2 3 1 2 3 1 3 1 2 3 1 2 3 1 2 1 2 3 1 2 3 1 2 3 2 3 1 2 3 1 2 3 1
35:     2 1 2 3 3 1 2 3 1 2 3 1 2 1 2 3 1 2 3 1 2 3 3 1 2 3 1 2 3 1 2 2 3 1 2 3 1 2 3 1
36:     2 1 2 3 3 1 2 3 1 2 3 1 2 2 3 1 2 3 1 2 3 1 1 2 3 1 2 3 1 2 3 3 1 2 3 1 2 3 1 2
37:     2 1 2 3 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 2 3 1 2 3 1 2 3 1 1 2 3 1 2 3 1 2 3
38:     2 2 3 1 1 2 3 2 3 1 3 1 2 1 2 3 2 3 1 3 1 2 1 2 3 2 3 1 3 1 2 1 2 3 2 3 1 3 1 2
39:     2 2 3 1 1 2 3 2 3 1 3 1 2 2 3 1 3 1 2 1 2 3 2 3 1 3 1 2 1 2 3 2 3 1 3 1 2 1 2 3
40:     2 2 3 1 1 2 3 2 3 1 3 1 2 3 1 2 1 2 3 2 3 1 3 1 2 1 2 3 2 3 1 3 1 2 1 2 3 2 3 1
41:     2 2 3 1 2 3 1 3 1 2 1 2 3 1 2 3 2 3 1 3 1 2 2 3 1 3 1 2 1 2 3 3 1 2 1 2 3 2 3 1
42:     2 2 3 1 2 3 1 3 1 2 1 2 3 2 3 1 3 1 2 1 2 3 3 1 2 1 2 3 2 3 1 1 2 3 2 3 1 3 1 2
43:     2 2 3 1 2 3 1 3 1 2 1 2 3 3 1 2 1 2 3 2 3 1 1 2 3 2 3 1 3 1 2 2 3 1 3 1 2 1 2 3
44:     2 2 3 1 3 1 2 1 2 3 2 3 1 1 2 3 2 3 1 3 1 2 3 1 2 1 2 3 2 3 1 2 3 1 3 1 2 1 2 3
45:     2 2 3 1 3 1 2 1 2 3 2 3 1 2 3 1 3 1 2 1 2 3 1 2 3 2 3 1 3 1 2 3 1 2 1 2 3 2 3 1
46:     2 2 3 1 3 1 2 1 2 3 2 3 1 3 1 2 1 2 3 2 3 1 2 3 1 3 1 2 1 2 3 1 2 3 2 3 1 3 1 2
47:     2 3 1 2 1 2 3 3 1 2 2 3 1 1 2 3 3 1 2 2 3 1 1 2 3 3 1 2 2 3 1 1 2 3 3 1 2 2 3 1
48:     2 3 1 2 1 2 3 3 1 2 2 3 1 2 3 1 1 2 3 3 1 2 2 3 1 1 2 3 3 1 2 2 3 1 1 2 3 3 1 2
49:     2 3 1 2 1 2 3 3 1 2 2 3 1 3 1 2 2 3 1 1 2 3 3 1 2 2 3 1 1 2 3 3 1 2 2 3 1 1 2 3
50:     2 3 1 2 2 3 1 1 2 3 3 1 2 1 2 3 3 1 2 2 3 1 2 3 1 1 2 3 3 1 2 3 1 2 2 3 1 1 2 3
51:     2 3 1 2 2 3 1 1 2 3 3 1 2 2 3 1 1 2 3 3 1 2 3 1 2 2 3 1 1 2 3 1 2 3 3 1 2 2 3 1
52:     2 3 1 2 2 3 1 1 2 3 3 1 2 3 1 2 2 3 1 1 2 3 1 2 3 3 1 2 2 3 1 2 3 1 1 2 3 3 1 2
53:     2 3 1 2 3 1 2 2 3 1 1 2 3 1 2 3 3 1 2 2 3 1 3 1 2 2 3 1 1 2 3 2 3 1 1 2 3 3 1 2
54:     2 3 1 2 3 1 2 2 3 1 1 2 3 2 3 1 1 2 3 3 1 2 1 2 3 3 1 2 2 3 1 3 1 2 2 3 1 1 2 3
55:     2 3 1 2 3 1 2 2 3 1 1 2 3 3 1 2 2 3 1 1 2 3 2 3 1 1 2 3 3 1 2 1 2 3 3 1 2 2 3 1
56:     3 1 3 2 1 3 2 1 3 2 1 3 2 1 3 2 1 3 2 1 3 2 1 3 2 1 3 2 1 3 2 1 3 2 1 3 2 1 3 2
57:     3 1 3 2 1 3 2 1 3 2 1 3 2 2 1 3 2 1 3 2 1 3 2 1 3 2 1 3 2 1 3 2 1 3 2 1 3 2 1 3
58:     3 1 3 2 1 3 2 1 3 2 1 3 2 3 2 1 3 2 1 3 2 1 3 2 1 3 2 1 3 2 1 3 2 1 3 2 1 3 2 1
59:     3 1 3 2 2 1 3 2 1 3 2 1 3 1 3 2 1 3 2 1 3 2 2 1 3 2 1 3 2 1 3 3 2 1 3 2 1 3 2 1
60:     3 1 3 2 2 1 3 2 1 3 2 1 3 2 1 3 2 1 3 2 1 3 3 2 1 3 2 1 3 2 1 1 3 2 1 3 2 1 3 2
61:     3 1 3 2 2 1 3 2 1 3 2 1 3 3 2 1 3 2 1 3 2 1 1 3 2 1 3 2 1 3 2 2 1 3 2 1 3 2 1 3
62:     3 1 3 2 3 2 1 3 2 1 3 2 1 1 3 2 1 3 2 1 3 2 3 2 1 3 2 1 3 2 1 2 1 3 2 1 3 2 1 3
63:     3 1 3 2 3 2 1 3 2 1 3 2 1 2 1 3 2 1 3 2 1 3 1 3 2 1 3 2 1 3 2 3 2 1 3 2 1 3 2 1
64:     3 1 3 2 3 2 1 3 2 1 3 2 1 3 2 1 3 2 1 2 1 3 2 1 3 2 1 3 1 3 2 1 3 2 1 3 2 1 3 2
65:     3 2 1 3 1 3 2 2 1 3 3 2 1 1 3 2 2 1 3 3 2 1 1 3 2 2 1 3 3 2 1 1 3 2 2 1 3 3 2 1
66:     3 2 1 3 1 3 2 2 1 3 3 2 1 2 1 3 3 2 1 1 3 2 2 1 3 3 2 1 1 3 2 2 1 3 3 2 1 1 3 2
67:     3 2 1 3 1 3 2 2 1 3 3 2 1 3 2 1 1 3 2 2 1 3 3 2 1 1 3 2 2 1 3 3 2 1 1 3 2 2 1 3
68:     3 2 1 3 2 1 3 3 2 1 1 3 2 1 3 2 2 1 3 3 2 1 2 1 3 3 2 1 1 3 2 3 2 1 1 3 2 2 1 3
69:     3 2 1 3 2 1 3 3 2 1 1 3 2 2 1 3 3 2 1 1 3 2 3 2 1 1 3 2 2 1 3 1 3 2 2 1 3 3 2 1
70:     3 2 1 3 2 1 3 3 2 1 1 3 2 3 2 1 1 3 2 2 1 3 1 3 2 2 1 3 3 2 1 2 1 3 3 2 1 1 3 2
71:     3 2 1 3 3 2 1 1 3 2 2 1 3 1 3 2 2 1 3 3 2 1 3 2 1 1 3 2 2 1 3 2 1 3 3 2 1 1 3 2
72:     3 2 1 3 3 2 1 1 3 2 2 1 3 2 1 3 3 2 1 1 3 2 1 3 2 2 1 3 3 2 1 3 2 1 1 3 2 2 1 3
73:     3 2 1 3 3 2 1 1 3 2 2 1 3 3 2 1 1 3 2 2 1 3 2 1 3 3 2 1 1 3 2 1 3 2 2 1 3 3 2 1
74:     3 3 2 1 1 3 2 3 2 1 2 1 3 1 3 2 3 2 1 2 1 3 1 3 2 3 2 1 2 1 3 1 3 2 3 2 1 2 1 3
75:     3 3 2 1 1 3 2 3 2 1 2 1 3 2 1 3 1 3 2 3 2 1 2 1 3 1 3 2 3 2 1 2 1 3 1 3 2 3 2 1
76:     3 3 2 1 1 3 2 3 2 1 2 1 3 3 2 1 2 1 3 1 3 2 3 2 1 2 1 3 1 3 2 3 2 1 2 1 3 1 3 2
77:     3 3 2 1 2 1 3 1 3 2 3 2 1 1 3 2 3 2 1 2 1 3 2 1 3 1 3 2 3 2 1 3 2 1 2 1 3 1 3 2
78:     3 3 2 1 2 1 3 1 3 2 3 2 1 2 1 3 1 3 2 3 2 1 3 2 1 2 1 3 1 3 2 1 3 2 3 2 1 2 1 3
79:     3 3 2 1 2 1 3 1 3 2 3 2 1 3 2 1 2 1 3 1 3 2 1 3 2 3 2 1 2 1 3 2 1 3 1 3 2 3 2 1
80:     3 3 2 1 3 2 1 2 1 3 1 3 2 1 3 2 3 2 1 2 1 3 3 2 1 2 1 3 1 3 2 2 1 3 1 3 2 3 2 1
81:     3 3 2 1 3 2 1 2 1 3 1 3 2 2 1 3 1 3 2 3 2 1 1 3 2 3 2 1 2 1 3 3 2 1 2 1 3 1 3 2
82:     3 3 2 1 3 2 1 2 1 3 1 3 2 3 2 1 2 1 3 1 3 2 2 1 3 1 3 2 3 2 1 1 3 2 3 2 1 2 1 3];
```

Appendices G List of Standard Orthogonal Arrays

Table G.1 shows the standard orthogonal arrays that exist for MDOE. The number of runs required are given by the subscript of the array name, and the number of factors at each level specified is given.

Table G.1: Standard orthogonal arrays

Orthogonal Array	**Number of Rows**	**Maximum Factors**	**Levels**			
			2	**3**	**4**	**5**
L_4	4	3	3	–	–	–
L_8	8	7	7	–	–	–
L_9	9	4	–	4	–	–
L_{12}	12	11	11	–	–	–
L_{16}	16	15	15	–	–	–
L'_{16}	16	5	–	–	5	–
L_{18}	18	8	1	7	–	–
L_{25}	25	6	–	–	–	6
L_{27}	27	13	–	13	–	–
L_{32}	32	31	31	–	–	–
L'_{32}	32	10	1	–	–	–
L_{36}	36	23	11	12	–	–
L'_{36}	36	16	3	13	–	–
L_{50}	50	12	1	–	–	11
L_{54}	54	26	1	25	–	–
L'_{64}	64	63	63	–	–	–
L_{64}	64	21	–	–	21	–
L'_{81}	81	40	–	40	–	–

Appendices H L_{81} Example Analysis

To ensure that the fractional factorial analysis undertaken in §9.4 was free of error, a known example was set up using the same column assignments as those used later for the fractional-factorial optimisation procedure, detailed in Table 9.6. Here a full quadratic 6-factor equation was devised, as shown in equation H.1. Using the same column assignments means the same test matrix was used, and responses were calculated for each run. The accuracy of the result will reveal any problems with the test matrix, such as confounding or other related issues.

$$\begin{aligned} R = {} & 4 + 8x_1 + 14x_2 + 6x_3 - 9x_4 - 13x_5 - 8x_6 + 12x_1x_2 - 6x_1x_3 \\ & + 14x_1x_4 - 2x_1x_5 - 7x_1x_6 - 1x_2x_3 - 4x_2x_4 + 7x_2x_5 + 10x_2x_6 \\ & - 12x_3x_4 - 11x_3x_5 + 3x_3x_6 - 3x_4x_5 + 2x_4x_6 + 11x_5x_6 \\ & + x_1^2 - 10x_2^2 - 5x_3^2 + 13x_4^2 + 5x_5^2 + 9x_6^2 \end{aligned} \tag{H.1}$$

The test matrix and calculated response appears in Table H.1. Column headers prefixed by "C" indicate column numbers from the original L_{81} array.

Table H.1: Factor levels and responses for fractional-factorial check

C1	C2	C5	C9	C14	C18	*R*
1	1	1	1	1	1	28
1	1	1	1	2	2	81
1	1	1	1	3	3	184
1	1	2	2	1	1	7
1	1	2	2	2	2	51
1	1	2	2	3	3	145
1	1	3	3	1	1	-22
1	1	3	3	2	2	13
1	1	3	3	3	3	98
1	2	1	2	1	2	109
1	2	1	2	2	3	207
1	2	1	2	3	1	85
1	2	2	3	1	2	102

Continued on next page...

Table H.1: ... continued

C1	C2	C5	C9	C14	C18	*R*
1	2	2	3	2	3	191
1	2	2	3	3	1	45
1	2	3	1	1	2	6
1	2	3	1	2	3	89
1	2	3	1	3	1	-60
1	3	1	3	1	3	230
1	3	1	3	2	1	100
1	3	1	3	3	2	206
1	3	2	1	1	3	126
1	3	2	1	2	1	-7
1	3	2	1	3	2	93
1	3	3	2	1	3	83
1	3	3	2	2	1	-74
1	3	3	2	3	2	17
2	1	1	2	1	2	124
2	1	1	2	2	3	196
2	1	1	2	3	1	99
2	1	2	3	1	2	130
2	1	2	3	2	3	193
2	1	2	3	3	1	72
2	1	3	1	1	2	-7
2	1	3	1	2	3	50
2	1	3	1	3	1	-74
2	2	1	3	1	3	278
2	2	1	3	2	1	173
2	2	1	3	3	2	253
2	2	2	1	1	3	133
2	2	2	1	2	1	25
2	2	2	1	3	2	99
2	2	3	2	1	3	103
2	2	3	2	2	1	-29
2	2	3	2	3	2	36
2	3	1	1	1	1	70
2	3	1	1	2	2	148
2	3	1	1	3	3	276
2	3	2	2	1	1	47
2	3	2	2	2	2	116
2	3	2	2	3	3	235
2	3	3	3	1	1	16

Continued on next page...

Table H.1: ... continued

C1	C2	C5	C9	C14	C18	*R*
2	3	3	3	2	2	76
2	3	3	3	3	3	186
3	1	1	3	1	3	284
3	1	1	3	2	1	204
3	1	1	3	3	2	258
3	1	2	1	1	3	98
3	1	2	1	2	1	15
3	1	2	1	3	2	63
3	1	3	2	1	3	81
3	1	3	2	2	1	-26
3	1	3	2	3	2	13
3	2	1	1	1	1	106
3	2	1	1	2	2	158
3	2	1	1	3	3	260
3	2	2	2	1	1	96
3	2	2	2	2	2	139
3	2	2	2	3	3	232
3	2	3	3	1	1	78
3	2	3	3	2	2	112
3	2	3	3	3	3	196
3	3	1	2	1	2	211
3	3	1	2	2	3	308
3	3	1	2	3	1	197
3	3	2	3	1	2	215
3	3	2	3	2	3	303
3	3	2	3	3	1	168
3	3	3	1	1	2	58
3	3	3	1	2	3	140
3	3	3	1	3	1	2

Running the data through an analysis similar to that described in §9.3 (selecting Quadratic from the Process Order menu) yielded the result shown in Fig. H.1. It can be seen by comparison with equation H.1 that the coefficients of each term were calculated correctly. We can therefore be confident of the results of the analysis in §9.4.

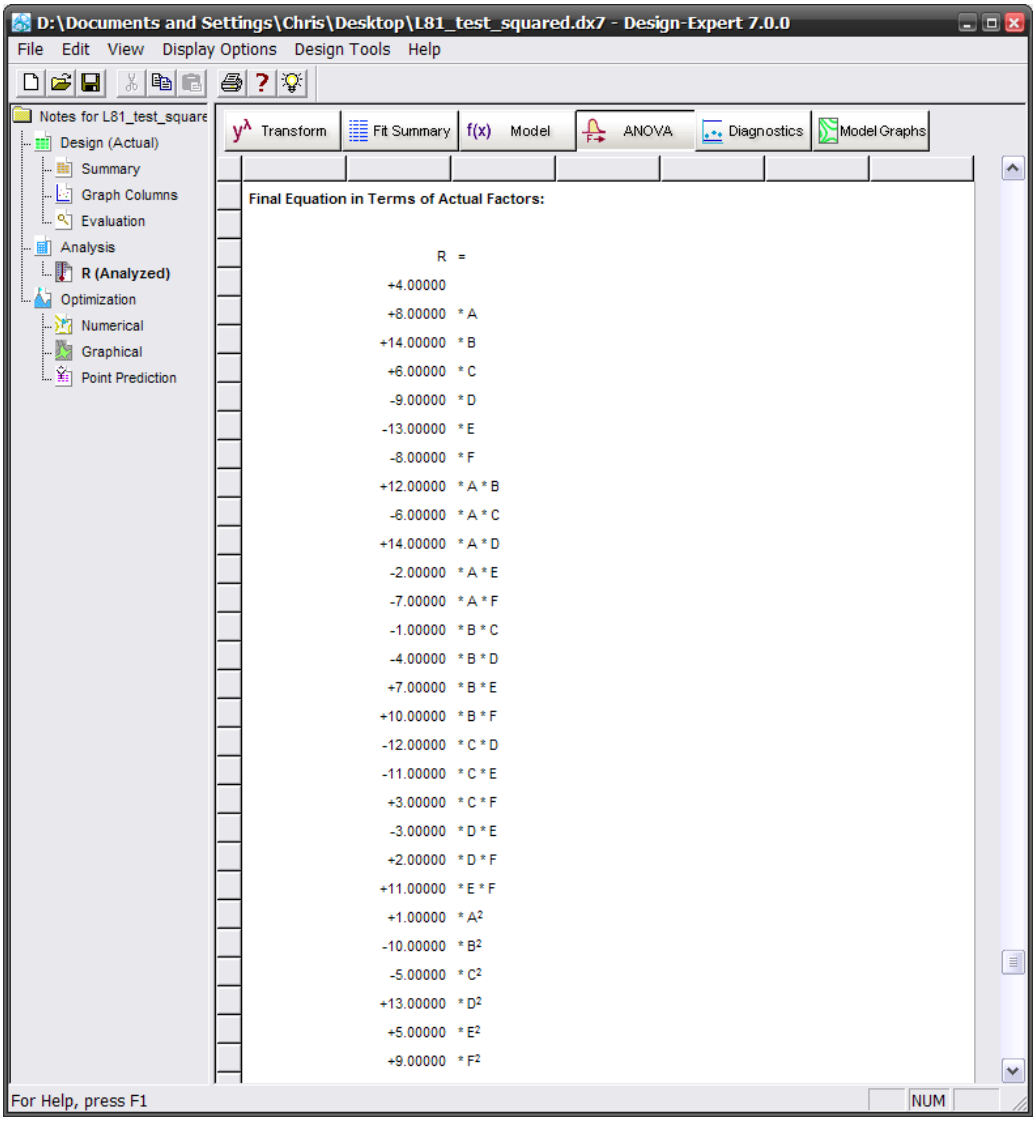


Figure H.1: Final equation as calculated by Design Expert

Appendices I Merit Function Analysis Code

The merit function was calculated for each analysis as described in §10.4.1. Below is the code for the fractional-factorial analysis, with the full-factorial analysis identical except for the equations used. Each airfoil shape has its own set of equations, and the calculation for each airfoil is identical except for the term coefficients. Only the NACA 00012 has therefore been included.

An explanation of the program follows the code.

I.1 `merit_function_L81.m`

Factors levels were set between the tested ranges, with 8 levels to show how the variation of each factor affects the merit function, as seen in Figs. 10.26 and 10.27. The maximum range of δ_t is set by the variable `max_angle`, as explained in §10.4.2.

All factors are given an even number of levels (12–15). If an odd number is used δ_t will be zero for the case where it should be non-zero. The maximum angle of $|\delta_t|$ is limited to 2.737, as explained in §10.4.2 (10).

For all combinations of NACA, $x_{h,\,a}$, $x_{h,\,t}$, α, and δ_t, δ_a is calculated so that $C_{H_a} = 0$ (32-58). δ_a is then calculated for $\delta_t = 0$ so that $C_{H_a} = 0$ (60–80)[1]. Using these values, C_L is calculated for $\delta_t = 0$ (82–90) and $\delta_t \neq 0$ (92–104). Finally C_{H_t} is calculated for $\delta_t \neq 0$ (106–119).

The merit function is recorded for each factor combination along with the factor levels themselves (301–302). The average values of each factor level are calculated (311–315) and plotted to give the results as they appear in Figs. 10.26 and 10.27.

```
1:    % This script file calculates the merit function for a range of
2:    % factor levels, and plots the factor effects.
3:
4:    clc
5:    clear all
6:    close all
7:
8:    % Set up variables to store merit function for all configurations
9:    k = 0;
10:   max_angle = 2.737;
11:   levels = 8;  % must be an even number
12:   xa = linspace(0.74,0.78,levels);
13:   xt = linspace(0.9324,0.9516,levels);
14:   alfa = linspace(-4,4,levels);
```

[1] The larger value of δ_a was chosen, as is explained in §I.2.

```
15:     deltat = linspace(-max_angle,max_angle,levels);
16:     merit = zeros(3*levels^4,6);
17:     m = 0;
18:     for x1 = 1:3
19:       for x2 = 1:levels
20:           for x3 = 1:levels
21:              for x4 = 1:levels
22:                 for x5 = 1:levels
23:                    m = m+1;
24:                    naca = x1;
25:                    x_a = xa(x2);
26:                    x_t = xt(x3);
27:                    alpha = alfa(x4);
28:                    delta_t = deltat(x5);
29:                    % 1: calculate delta_a for delta_t =/= 0, Cha = 0
30:                    switch naca
31:                       case 1 % NACA 00012
32:                          % 1: Find delta_a for Ch_a = 0, delta_t as above
33:                          % coefficient of delta_a^2
34:                          A = -1.0256849583263E-04;
35:
36:                          % coefficient of delta_a^1
37:                          B = -7.2071485776746E-03 ...
38:                              -4.5939610983112E-05 * alpha ...
39:                              -1.0018669296334E-04 * delta_t ...
40:                              +8.3740253462087E-03 * x_a;
41:
42:                          % coefficient of delta_a^0
43:                          C = -1.2670760066667E-02 ...
44:                             -4.7511342378174E-03 * alpha...
45:                             -1.4527111639032E-02 * delta_t...
46:                             +1.2536480193705E-02 * x_a...
47:                             +5.0456115708879E-03 * x_t...
48:                             -1.2445452047505E-05 * alpha * delta_t...
49:                             +5.6398520074425E-03 * alpha * x_a...
50:                             +9.7125217680697E-03 * delta_t * x_a...
51:                             +6.3182282485321E-03 * delta_t * x_t...
52:                             +6.5305475559740E-06 * alpha^2 ...
53:                             -2.6948531550324E-05 * delta_t^2;
54:                          if B^2-4*A*C<0
55:                             delta_a = NaN;
56:                          else
57:                             delta_a = (-B-sqrt(B^2-4*A*C))/(2*A);
58:                          end
59:
60:                          % 2: Find delta_a for Ch_a = 0, delta_t = 0
61:                          % coefficient of delta_a^2
62:                          A = -1.0256849583263E-04;
63:
64:                          % coefficient of delta_a^1
65:                          B = -7.2071485776746E-03 ...
66:                              -4.5939610983112E-05 * alpha ...
67:                              +8.3740253462087E-03 * x_a;
68:
69:                          % coefficient of delta_a^0
70:                          C = -1.2670760066667E-02 ...
71:                             -4.7511342378174E-03 * alpha...
72:                             +1.2536480193705E-02 * x_a...
73:                             +5.0456115708879E-03 * x_t...
74:                             +5.6398520074425E-03 * alpha * x_a...
75:                             +6.5305475559740E-06 * alpha^2;
76:                          if B^2-4*A*C<0
77:                             delta_a0 = NaN;
78:                          else
79:                             delta_a0 = (-B-sqrt(B^2-4*A*C))/(2*A);
80:                          end
```

```
81:
82:                         % 3: Find Cl for delta_t = 0
83:                         Cl_0 = 4.0174649669300E-02 ...
84:                               +1.1631372561235E-01 * alpha...
85:                               +1.7823664346806E-01 * delta_a...
86:                               -1.1914712970573E-02 * x_a...
87:                               -3.2692979184961E-02 * x_t...
88:                               +4.7841896795207E-03 * alpha * x_a...
89:                               -1.3965513444496E-01 * delta_a * x_a...
90:                               -2.8817918326222E-05 * delta_a^2;
91:
92:                         % 4: Find Cl for delta_t =/= 0
93:                         Cl = 4.0174649669300E-02 ...
94:                            +1.1631372561235E-01 * alpha...
95:                            +1.7823664346806E-01 * delta_a...
96:                            +3.3203651450598E-01 * delta_t...
97:                            -1.1914712970573E-02 * x_a...
98:                            -3.2692979184961E-02 * x_t...
99:                            -1.6767694087078E-05 * alpha * delta_t...
100:                           +4.7841896795207E-03 * alpha * x_a...
101:                           -1.3965513444496E-01 * delta_a * x_a...
102:                           +6.9665921498745E-03 * delta_t * x_a...
103:                           -3.2041645265300E-01 * delta_t * x_t...
104:                           -2.8817918326222E-05 * delta_a^2;
105:
106:                        % 5: Find Ch_t for delta_t =/= 0
107:                        Ch_t = 7.2316927745756E-03 ...
108:                              -9.9659967474709E-06 * alpha...
109:                              -1.0041565949802E-03 * delta_a...
110:                              -1.7711466808057E-03 * delta_t...
111:                              -6.1457459696502E-03 * x_t...
112:                              -1.9776251167043E-06 * alpha * delta_a...
113:                              -3.7338227940557E-06 * alpha * delta_t...
114:                              -2.7414949620772E-05 * delta_a * delta_t...
115:                              +1.0554124007258E-03 * delta_a * x_t...
116:                              +1.8343971449765E-03 * delta_t * x_t...
117:                              +2.3814861014898E-06 * alpha^2 ...
118:                              -6.8475266085551E-06 * delta_a^2 ...
119:                              -1.9204539307736E-05 * delta_t^2;
120:                    case 2 % NACA 23012
 :                       % Similarly as above for NACA 00012
210:                    case 3 % NACA 23018
 :                       % Similarly as above for NACA 00012
300:                end
301:                % create array with factor levels and merit function
302:                merit(m,:) = [x1 x2 x3 x4 x5 (Cl - Cl_0)/Ch_t];
303:             end
304:          end
305:       end
306:    end
307: end
308:
309: %% Get factor effects from "merit" array
310: factor = NaN(5,levels);
311: for i = 1:5
312:    for j = 1:levels
313:       factor(i,j) = dot(merit(:,6),merit(:,i)==j)/sum(merit(:,i)==j);
314:    end
315: end
316:     colour = [1   0   0;        % red
317:               1   2/5 0;        % orange dark
318:               0   3/5 0;        % green
319:               0   2/5 4/5;      % blue
320:               0   0   3/5;      % dark blue
```

```
321:                    2/5 0   3/5];     % purple
322:
323:     % plot each factor effect
324:     hold on
325:     for i = 1:length(factor(:,1))
326:       plot((1:levels)+(i-1)*levels,factor(i,:),'o-','Color',colour(i,:)...
327:                                                    ,'LineWidth',2)
327:     end
```

I.2 Deciding which δ_a to use

It can be seen from the equations for C_{Ha} in §9.3.3 and §9.4.4 that for both analyses δ_a appears as a second-order term. This means that calculation of δ_a for $C_{Ha} = 0$ leads to a quadratic equation, which often yields two solutions, as seen in Fig. I.1 for a single configuration.

To choose the correct angle we see that at the smallest root of δ_a (labelled A in Fig. I.1) we have $\frac{\mathrm{d}C_{Ha}}{\mathrm{d}\delta_a} > 0$ and at the largest root (labelled B in Fig. I.1) we have $\frac{\mathrm{d}C_{Ha}}{\mathrm{d}\delta_a} < 0$. This is always the case, as the coefficient of δ_a^2 is negative for both analyses. Elementary small disturbance stability analysis indicates that A and B are unstable and stable equilibria, respectively.

The angle chosen then is the largest root, given by $\frac{b-\sqrt{b^2-4ac}}{2a}$ since a is always negative.

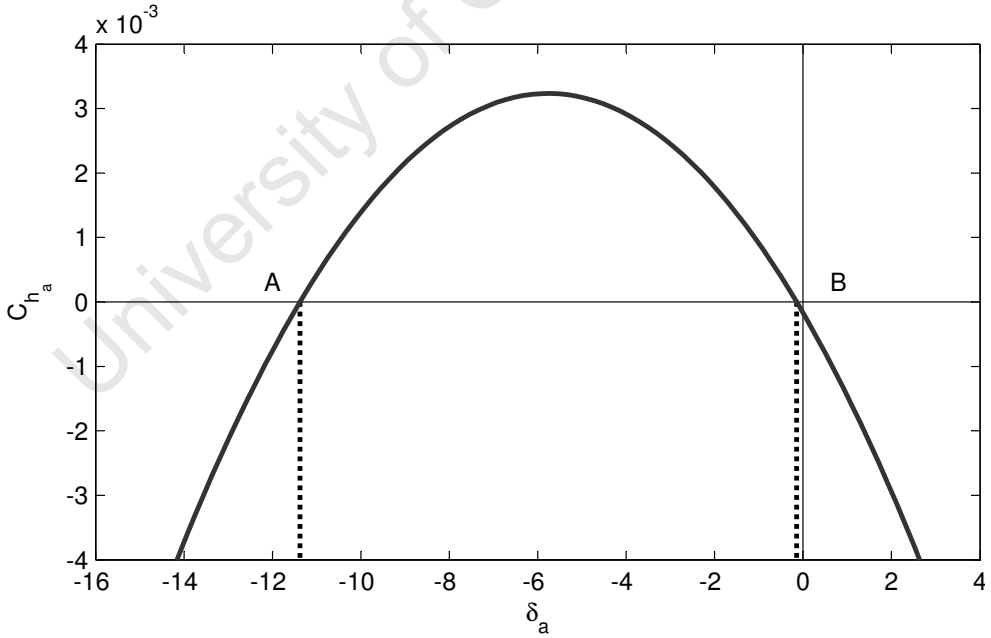


Figure I.1: Example of C_{Ha} vs. δ_a for a single configuration. Airfoil is a NACA 23018 with $x_{h,\,a} = 0.76$, $x_{h,\,t} = 0.9424$, $\alpha = -4°$, and $\delta_t = -4°$, using equations from the fractional-factorial analysis. Solutions exist at $\delta_a \approx -11.38$ and $\delta_a \approx -0.147$.

Appendices J Matlab Code for Interaction Plots

This appendix contains the Matlab code to plot the interactions that appear in §10.3. The code has been edited to remove some plotting features but the calculation of the plotted values stays intact.

The combinations that were plotted were chosen so that they best illustrated the interaction, each interaction having two options of which factor to plot on the abscissa.


```
% This function plots interactions for the full-factorial analysis of C_L.

% Written by Chris Day - 3 August 2011

clc
clear all
close all

% Load test array and responses
load('full_test_array_and_responses.mat')

% Factor levels as they appear in test array
alpha = [-4 0 4];  % factor 1
delta_a = [-4 0 4];  % factor 2
delta_t = [-4 0 4];  % factor 3
x_a = [0.74 0.76 0.78];  % factor 4
x_t = [0.9324 0.9424 0.9516];  % factor 5
airfoil = {'00012' '23012' '23018'};  % factor 6

% Plot interactions - Cl
for i = 1:6
    for j = 1:6
        a1 = test_matrix(:,i)==1;
        a2 = test_matrix(:,i)==2;
        a3 = test_matrix(:,i)==3;
        b1 = test_matrix(:,j)==1;
        b2 = test_matrix(:,j)==2;
        b3 = test_matrix(:,j)==3;
        a1b1 = dot(cl,a1.*b1)/sum(a1.*b1);
        a1b2 = dot(cl,a1.*b2)/sum(a1.*b2);
        a1b3 = dot(cl,a1.*b3)/sum(a1.*b3);
        a2b1 = dot(cl,a2.*b1)/sum(a2.*b1);
        a2b2 = dot(cl,a2.*b2)/sum(a2.*b2);
        a2b3 = dot(cl,a2.*b3)/sum(a2.*b3);
        a3b1 = dot(cl,a3.*b1)/sum(a3.*b1);
        a3b2 = dot(cl,a3.*b2)/sum(a3.*b2);
        a3b3 = dot(cl,a3.*b3)/sum(a3.*b3);
        figure
        hold on
        plot([1 2 3],[a1b3 a2b3 a3b3],'ro-','LineWidth',2)
        plot([1 2 3],[a1b2 a2b2 a3b2],'o-','LineWidth',2,'Color',[0 0.6 0])
        plot([1 2 3],[a1b1 a2b1 a3b1],'bo-','LineWidth',2)
```

```
43:         end
44:     end
```

Appendices K Matlab Code for Statistical Analysis

This appendix contains the Matlab code to plot the statistical outputs found in Figs. 10.28 to 10.45. Only the script for the fractional-factorial analysis is presented as only the equations differed between the analyses. In this script only the NACA 00012 equations have been included for brevity's sake, and only the plotting procedures for the analysis of C_L are included for the same reason.

The script first loads the test array and response recorded previously (11–13), and for each run first calculates the actual factor levels (25–30) then the response according to equations 9.12 to 9.20 (32–116). The residual and the standardised residual are then calculated (119–123) and several plots are output (125-15). This is repeated for both C_{H_a} and C_{H_t}.


```
1:   % This function calculates all responses from the full-factorial analysis
2:   % using equations from the fractional factorial analysis, in order to
3:   % evaluate the accuracy of the fractional-factorial experiment.
4:
5:   % Written by Chris Day - 3 August 2011
6:
7:   clc
8:   clear all
9:   close all
10:
11:  % Load test array and responses from full analysis
12:  load('full_test_array_and_responses.mat')
13:  OA = test_matrix;
14:
15:  % Factor levels as they appear in test array
16:  alpha_ = [-4 0 4];                    % factor 1
17:  deltaa = [-4 0 4];                    % factor 2
18:  deltat = [-4 0 4];                    % factor 3
19:  xa = [0.74 0.76 0.78];                % factor 4
20:  xt = [0.9324 0.9424 0.9516];          % factor 5
21:  airfoil = {'10012' '23012' '23018'};  % factor 6
22:
23:  % Calculate responses from L81 equations
24:  for i = 1:length(OA)
25:     alpha = alpha_(OA(i,1));
26:     delta_a = deltaa(OA(i,2));
27:     delta_t = deltat(OA(i,3));
28:     x_a = xa(OA(i,4));
29:     x_t = xt(OA(i,5));
30:     NACA(i) = airfoil(OA(i,6));
31:
32:     switch OA(i,6)
33:        case 1
34:           CL(i)  =  4.0174649669300E-02 + 1.1631372561235E-01 * alpha...
35:                   + 1.7823664346806E-01 * delta_a + 3.3203651450598E-01 ...
36:                   * delta_t - 1.1914712970573E-02 * x_a - 3.2692979184961E-02 ...
```

```
                    * x_t - 1.6767694087078E-05 * alpha * delta_t ...
                    + 4.7841896795207E-03 * alpha * x_a - 1.3965513444496E-01 ...
                    * delta_a * x_a + 6.9665921498745E-03 * delta_t * x_a ...
                    - 3.2041645265300E-01 * delta_t * x_t - 2.8817918326222E-05 ...
                    * delta_a^2;
            CHa(i) = -1.2670760066667E-02 - 4.7511342378174E-03 * alpha...
                    - 7.2071485776746E-03 * delta_a - 1.4527111639032E-02 ...
                    * delta_t + 1.2536480193705E-02 * x_a + 5.0456115708879E-03 ...
                    * x_t - 4.5939610983112E-05 * alpha * delta_a ...
                    - 1.2445452047505E-05 * alpha * delta_t + 5.6398520074425E-03 ...
                    * alpha * x_a - 1.0018669296334E-04 * delta_a * delta_t ...
                    + 8.3740253462087E-03 * delta_a * x_a + 9.7125217680697E-03 ...
                    * delta_t * x_a + 6.3182282485321E-03 * delta_t * x_t ...
                    + 6.5305475559740E-06 * alpha^2 - 1.0256849583263E-04 ...
                    * delta_a^2 - 2.6948531550324E-05 * delta_t^2;
            CHt(i) =  7.2316927745756E-03 - 9.9659967474709E-06 * alpha...
                    - 1.0041565949802E-03 * delta_a - 1.7711466808057E-03 ...
                    * delta_t - 6.1457459696502E-03 * x_t - 1.9776251167043E-06 ...
                    * alpha * delta_a - 3.7338227940557E-06 * alpha * delta_t ...
                    - 2.7414949620772E-05 * delta_a * delta_t ...
                    + 1.0554124007258E-03 * delta_a * x_t + 1.8343971449765E-03 ...
                    * delta_t * x_t + 2.3814861014898E-06 * alpha^2 ...
                    - 6.8475266085551E-06 * delta_a^2 - 1.9204539307736E-05 * delta_t^2;
        case 2
                      % Similarly as above for NACA 00012
        case 3
                      % Similarly as above for NACA 00012
    end
end

% cl ------------------------------------------------------------------------
e_cl = cl-CL';                                  % residual
n = length(OA);                                 % number of experimental runs
p = 12;                                         % columns of X matrix
standard_deviation = sqrt((e_cl'*e_cl)/(n-p));% Myers & Montgomery eq 2.18
d_cl = e_cl/standard_deviation;                 % standardised residual

% scatter plot of residual
figure
hold on
plot([-3 3],[3 3],'Color',[1 0.4 0])
plot([-3 3],[-3 -3],'Color',[1 0.4 0])
plot(CL,d_cl,'k.')
ylabel('Standardized Residual')
xlabel('Predicted C_L')
axis([-1.25 1.25 -4 4])

% normal probability plot
figure
normplot(d_cl)
ylabel('Normal % Probability')
xlabel('Standardized Residuals, d_i')

% histogram
figure
hist(e_cl,20)
ylabel('Frequency')
xlabel('Residuals, e_i')
```